Network Applications: Multi-Server Request Routing

Y. Richard Yang

http://zoo.cs.yale.edu/classes/cs433/

10/16/2018

Outline

- Admin and recap
- Multiple servers

Admin

- Assignment Three office hours this week
  - Wednesday: 1:30-2:30pm
  - Thursday: 1:30-2:30 pm
  - Friday: 1:00-2:00 pm
- Exam 1 date?

Recap: High-Performance Network Server

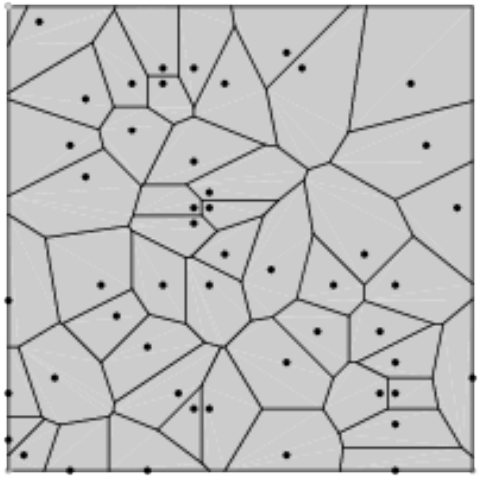
- Problem: avoid blocking (so that we can reach bottleneck throughput)
  - Introduce threads, async io
- Problem: limit unlimited thread overhead
  - Thread pool
- Problem: shared variables
  - Synchronization (lock, synchronized)
- Problem: avoid busy-wait
  - Wait/notify; Condition, FSM; asynchronous channel/Future/Handler
- Problem: extensibility/robustness
  - Language support/Design for interfaces
- Problem: system modeling and measurements
  - Queueing analysis, operational analysis

Recap: Designing Load-Balancing Multiple Servers

- Requirements/goals
  - naming abstraction, server load balancing, failure detection, access control filtering, priorities/QoS, request locality, transparent caching
- Components
  - Service/resource discovery (static, zookeeper, etcs, consul)
  - Health/state monitoring of servers/connecting networks
  - Load balancing mechanisms/algorithm
    - Also called request routing

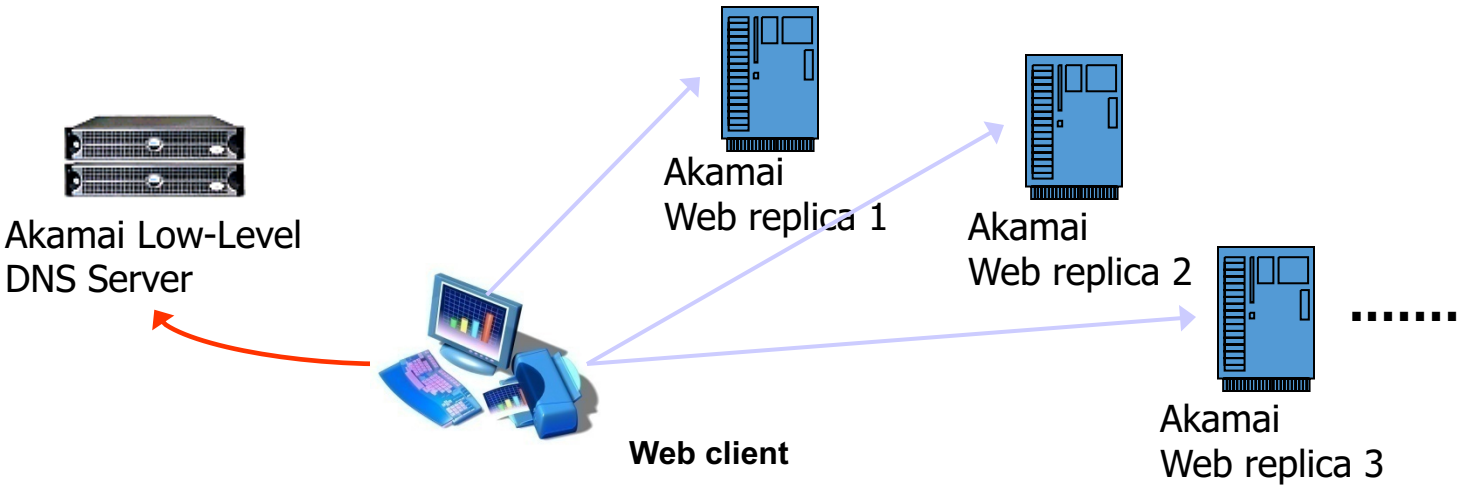
Recap: Load-Balancing/Request Routing using DNS

- Potential basic techniques (Akamai design)
  - One-level of indirection (aliasing, cname), e.g., dig results:
    - cdn.cnn.com. 56 IN CNAME ion-ma.turner.com.edgekey.net
    - ion-ma.turner.com.edgekey.net. 556 IN CNAME e12596.dscj.akamaiedge.net.
  - Hierarchy (multiple levels), e.g.,
    - first level: query dscj of name server akamaiedge.net. to decide region according to (dscj, clientIP)
    - next level: query e12596 of region name server to choose specific server



Experimental Study of Akamai Load Balancing

- Methodology
  - 2-months long measurement
  - 140 PlanetLab nodes (clients)
    - 50 US and Canada, 35 Europe, 18 Asia, 8 South America, the rest randomly scattered
  - Every 20 sec, each client queries an appropriate CNAME for Yahoo, CNN, Fox News, NY Times, etc.

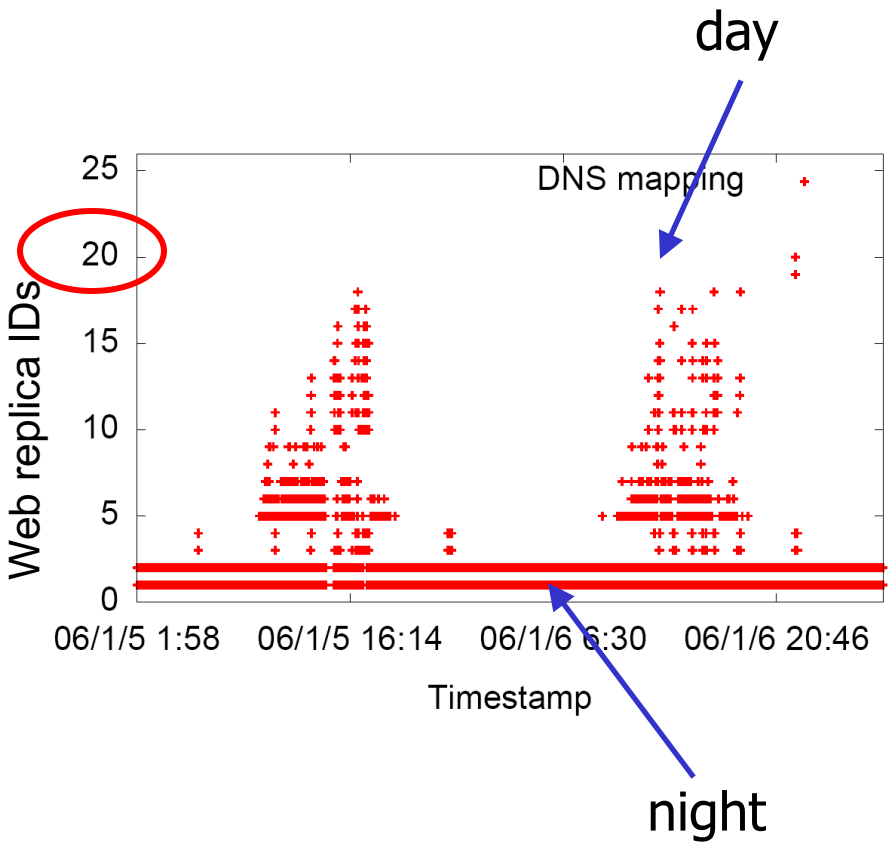


See http://www.aqualab.cs.northwestern.edu/publications/Ajsu06DBA.pdf

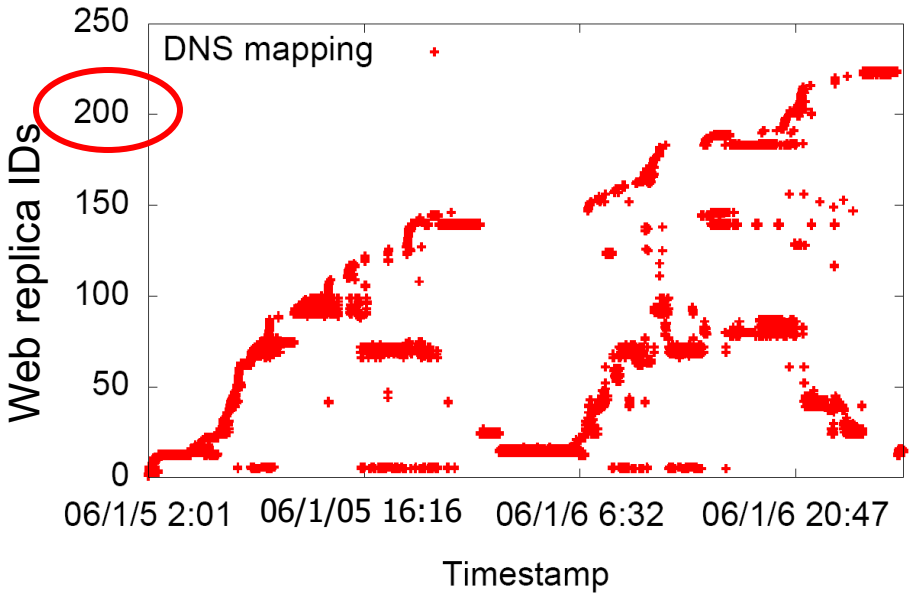
Server Pool: to Yahoo

Target: a943.x.a.yimg.com (Yahoo)

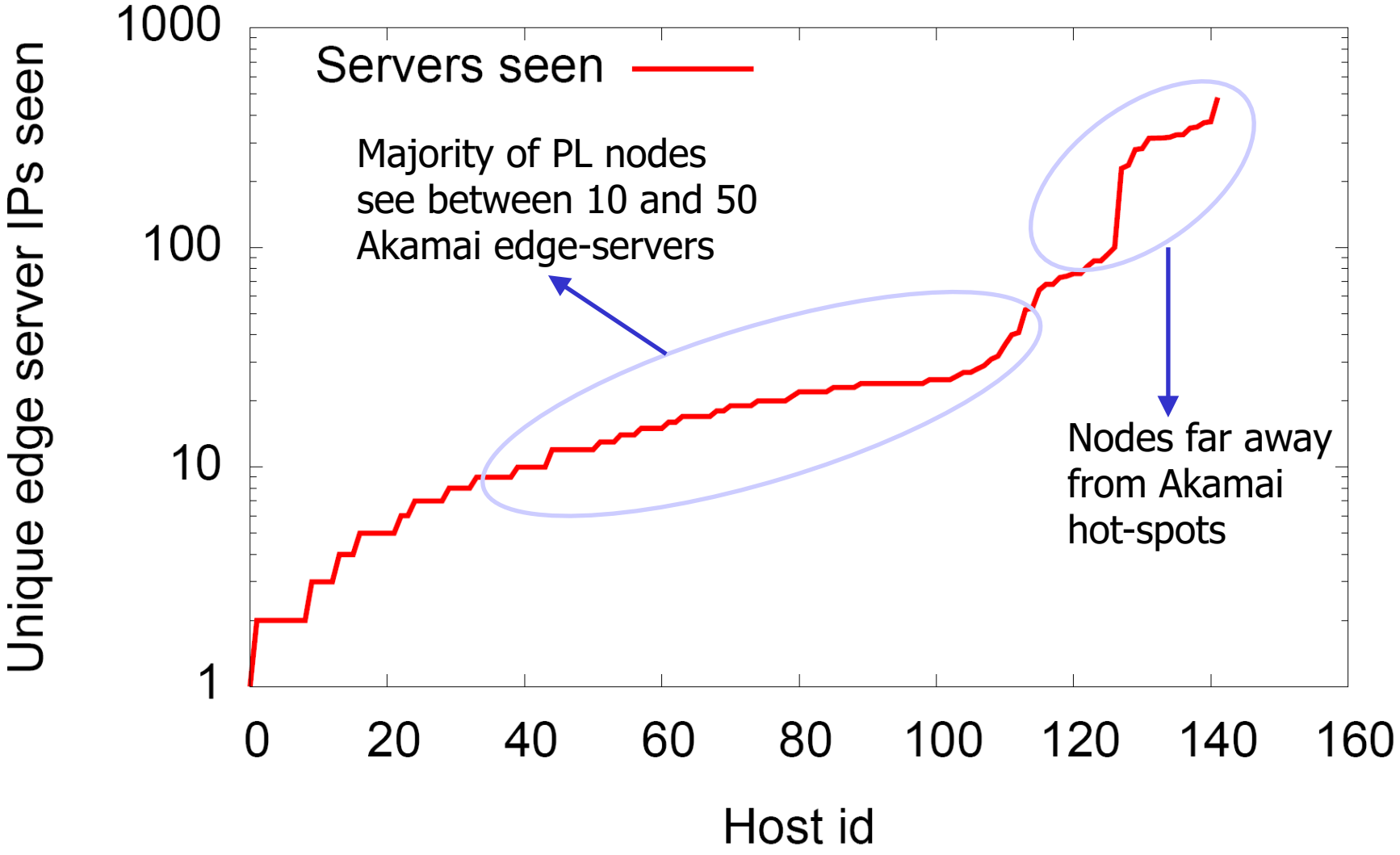
Client 1: Berkeley



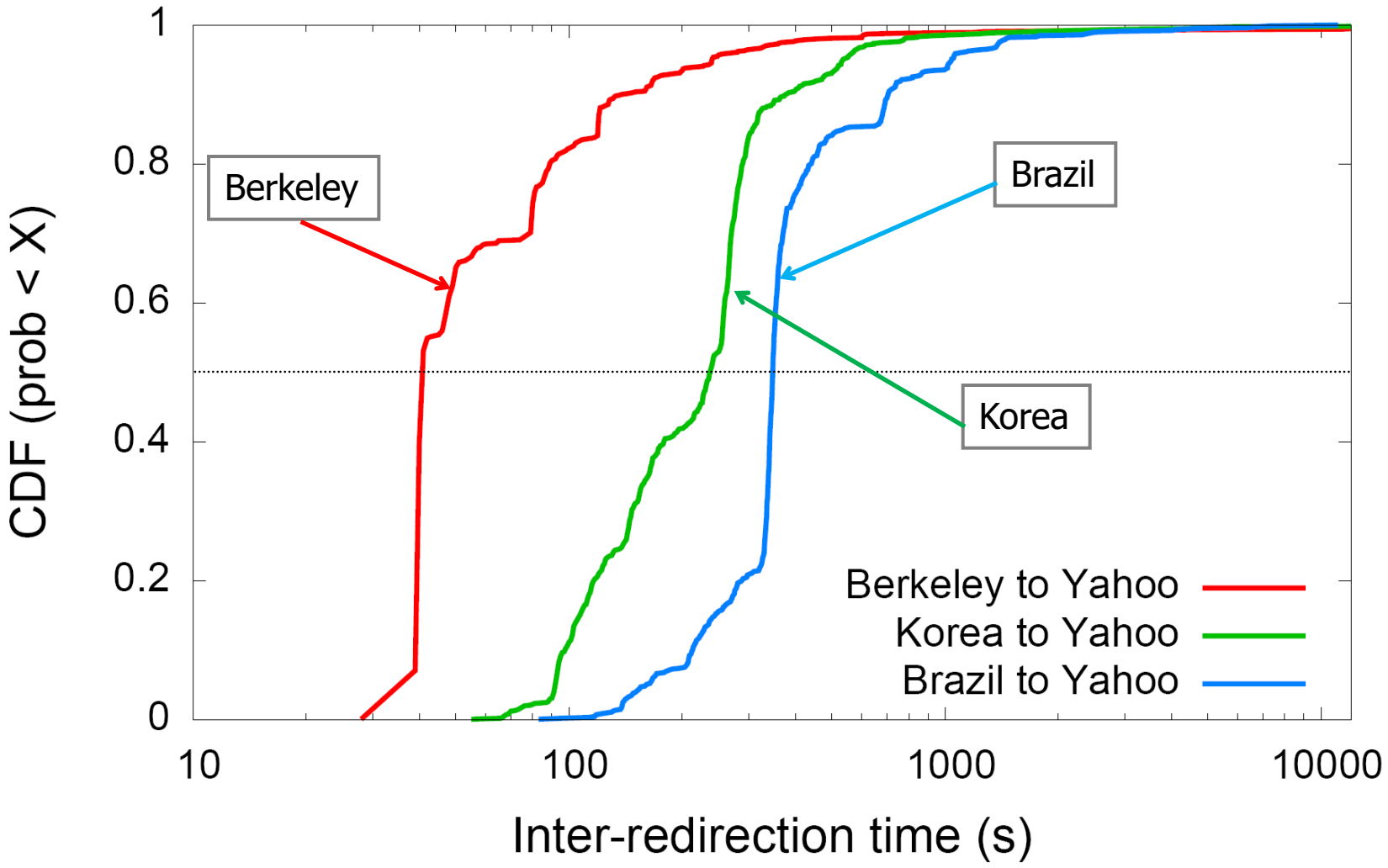
Client 2: Purdue



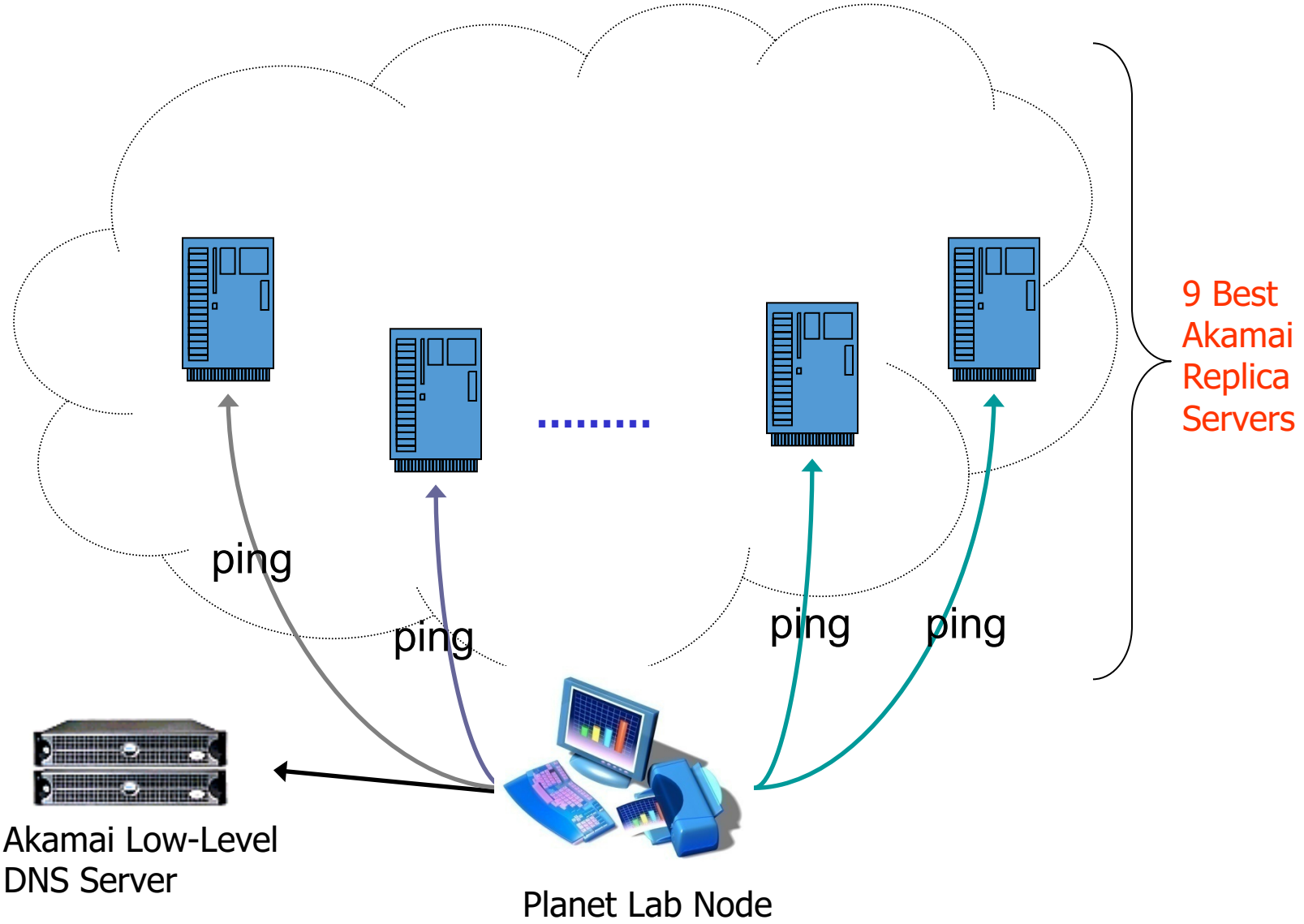
Server Diversity for Yahoo



Load Balancing Dynamics

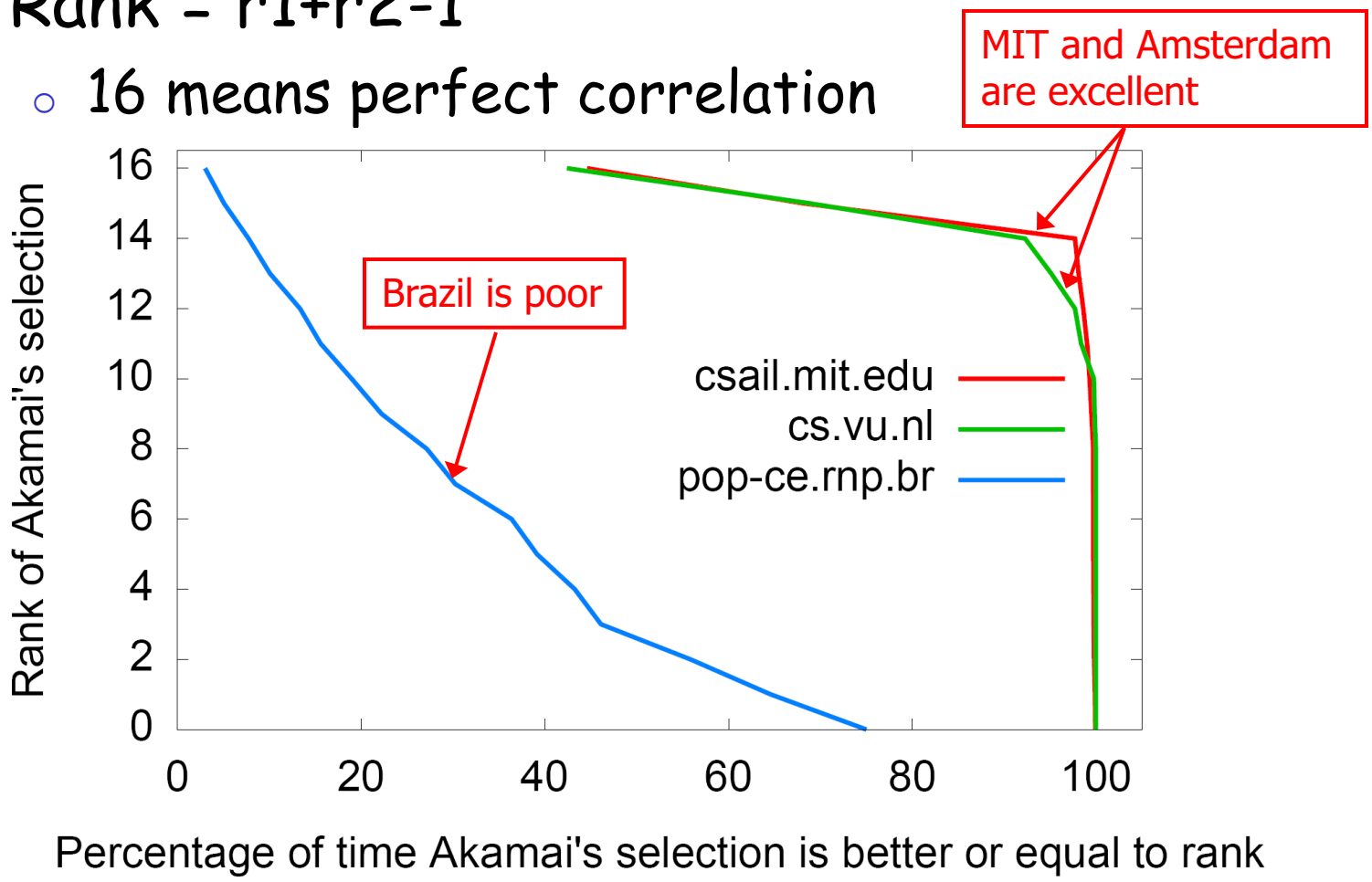


Redirection Effectiveness: Measurement Methodology



Do redirections reveal network conditions?

- Rank = r1+r2-1
  - 16 means perfect correlation



(Offline Read)

- Facebook DNS Load Direction
  - A system named Cartographer (written in Python) processes measurement data and configures the DNS maps of individual DNS servers (open source tinydns)
- Amazon AWS Route 53 service
  - https://aws.amazon.com/route53/

Discussion

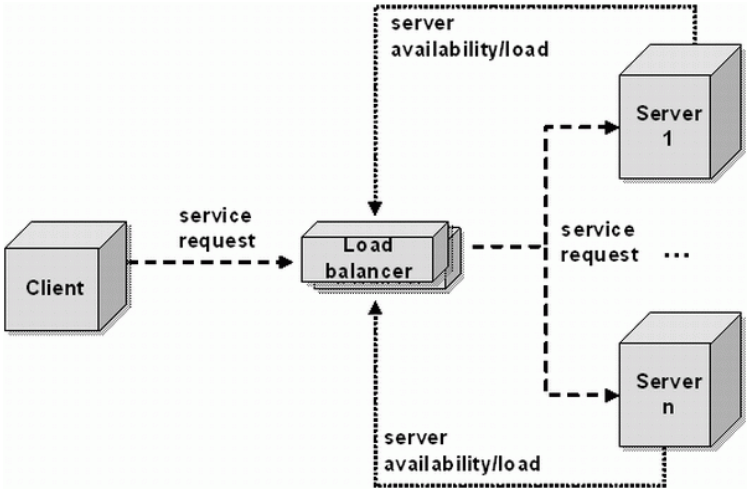
- Advantages and disadvantages of request routing using DNS

Outline

- Admin and recap
- Request routing to multiple servers
  - overview
  - DNS request routing
  - Network request routing

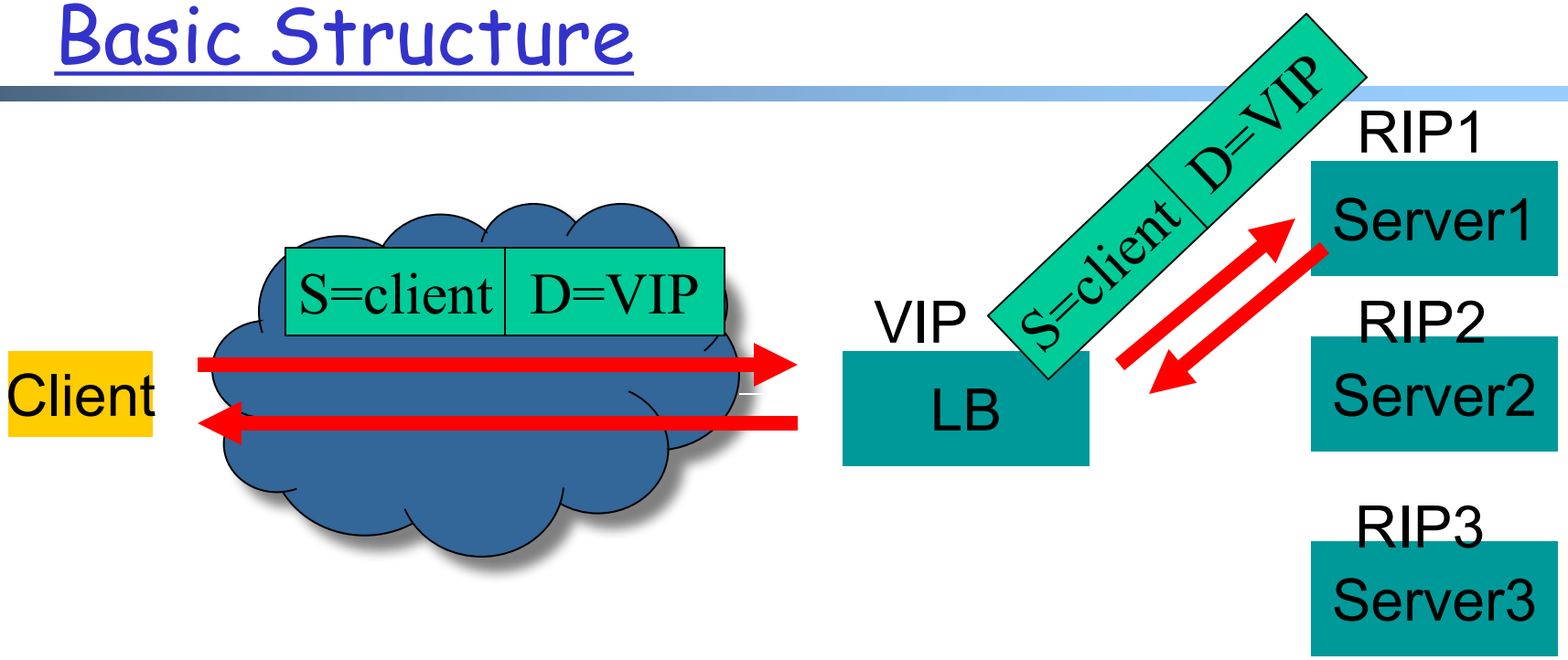
Network (L4) Request Routing: API

- A single service IP address (naming abstraction) can be used for a cluster of (physical) servers
  - Such a single IP is called a virtual IP address (VIP)



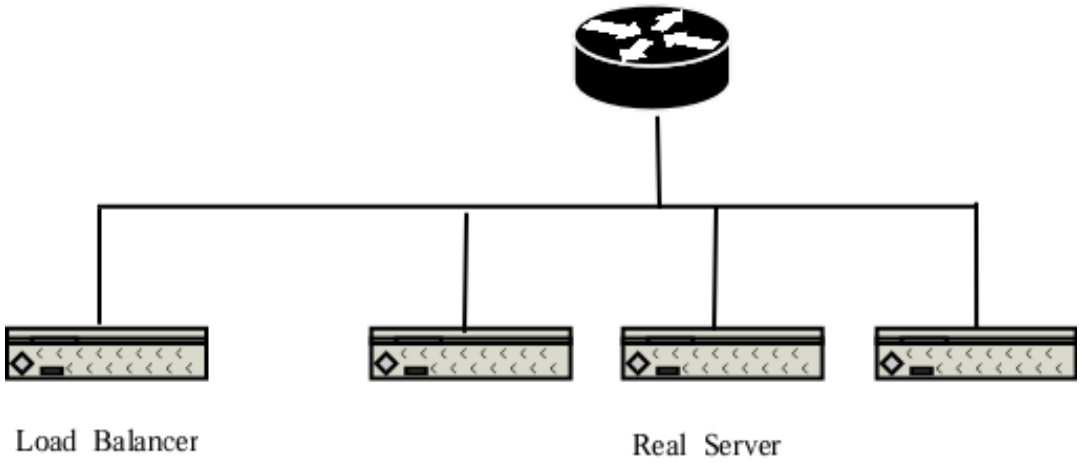

```
ipvsadm -A -t 192.168.0.1:80 -s rr
ipvsadm -a -t 192.168.0.1:80 -r 172.16.0.1:80 -m
ipvsadm -a -t 192.168.0.1:80 -r 172.16.0.2:80 -m
```

Network Load Balancing (NLB): Basic Structure



Problem: How can the LB send a request to chosen real server i?

Building Block: Layer 2 Forwarding and Address Resolution Protocol (ARP)



- Each network interface card listens to an assigned MAC address
- To send to a device with a given IP, the sender
  - first translates the IP of the destination to its MAC (device) address
    - The translation is done by the Address Resolution Protocol (ARP)
  - then sends the packet with the given MAC address
    - this is called layer 2 forwarding

ARP Protocol

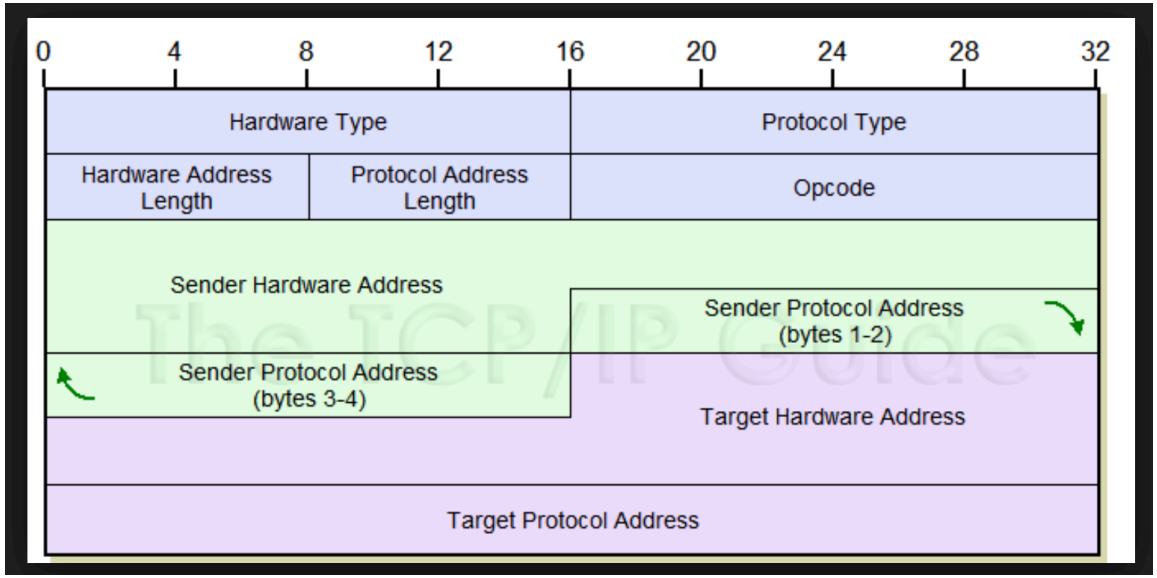
- ARP is "plug-and-play":
  - nodes create their ARP tables without intervention from net administrator

- A broadcast protocol:
  - Client broadcasts query frame, containing queried IP address
    - all machines on LAN receive ARP query
  - Node with queried IP receives ARP frame, replies its MAC address

Demo: ARP

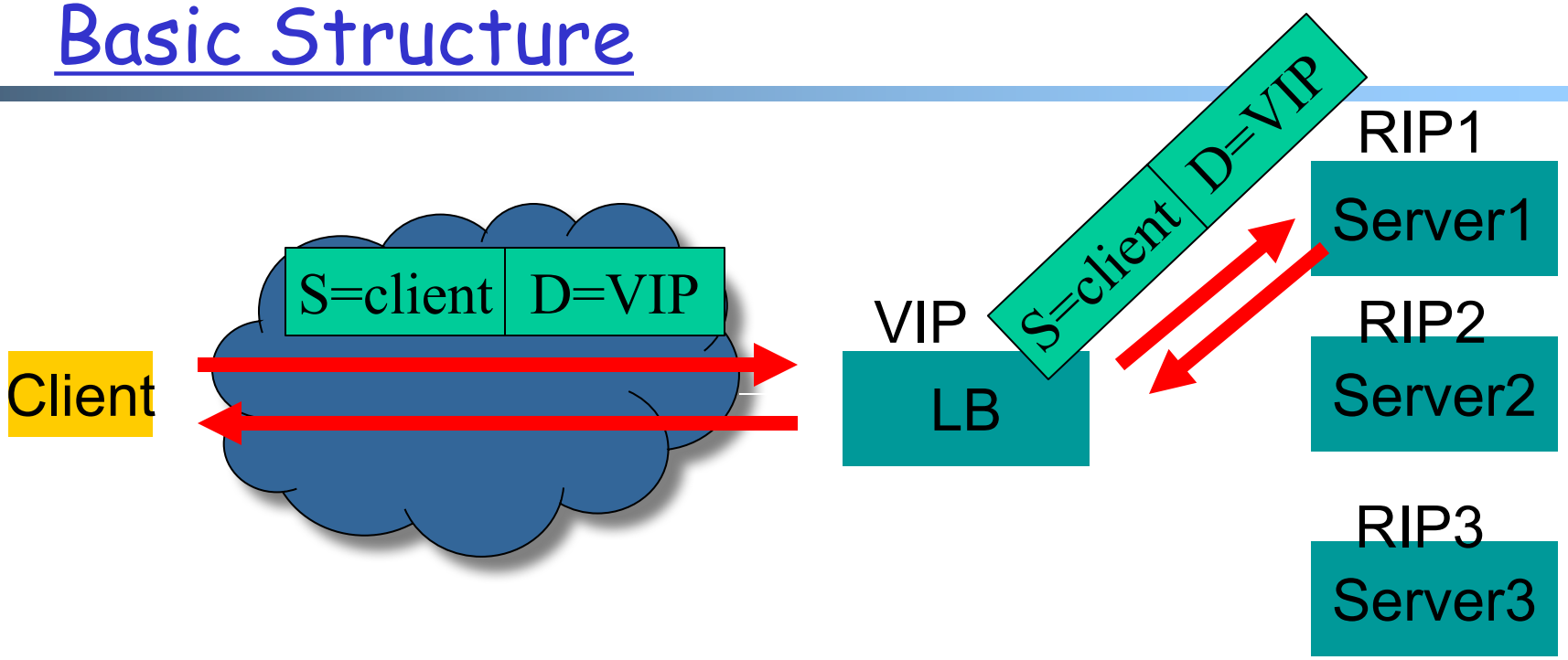
- ifconfig –a
  - to show all interfaces and their MAC addresses
- arp –a
  - show the binding between IP address and MAC address
- Wireshark to capture arp traffic

ARP Format and Features



- Query: Layer 2 (Link layer) broadcast: destination ff:ff:ff:ff:ff:ff to be received by all hosts at the same local network
- Response: Host with the MAC returns its MAC if it has the query IP
- Gratuitous ARP: A host sends this message to update other devices if it changes MAC

Network Load Balancing (NLB): Basic Structure

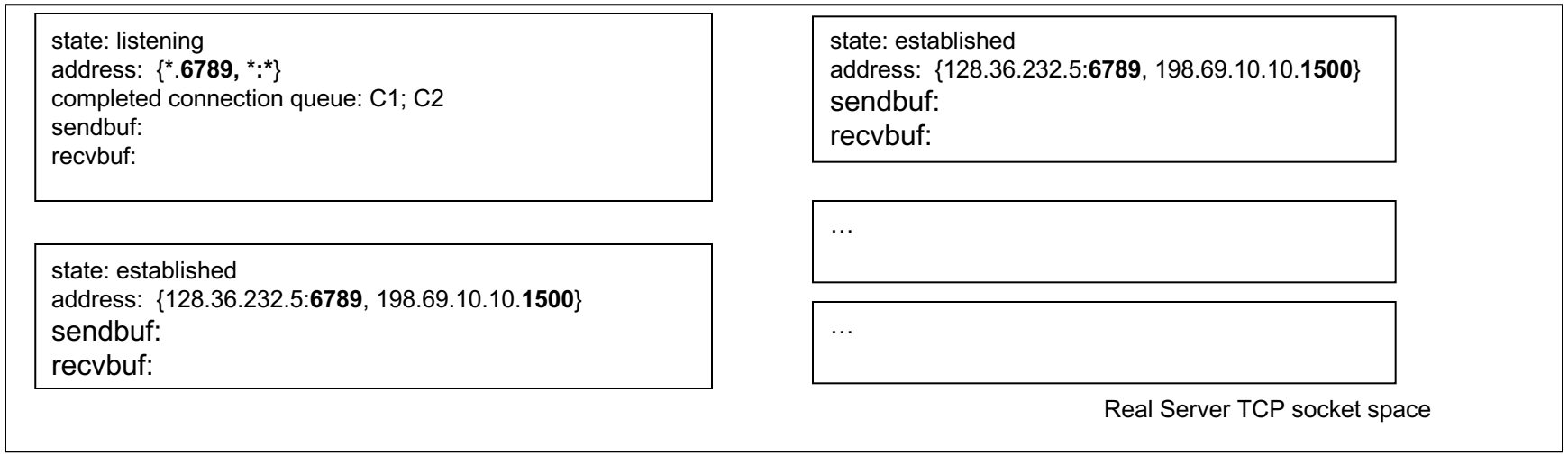


Problem of the basic structure?

Problem

S=client	D=VIP

- Although the request router can send to a real server, the packet has VIP as destination address, but a real server may use its own RIP
  - if NLB just forwards the packet from client to a real server, the real server drops the packet
  - reply from real server to client has real server IP as source -> client will drop the packet



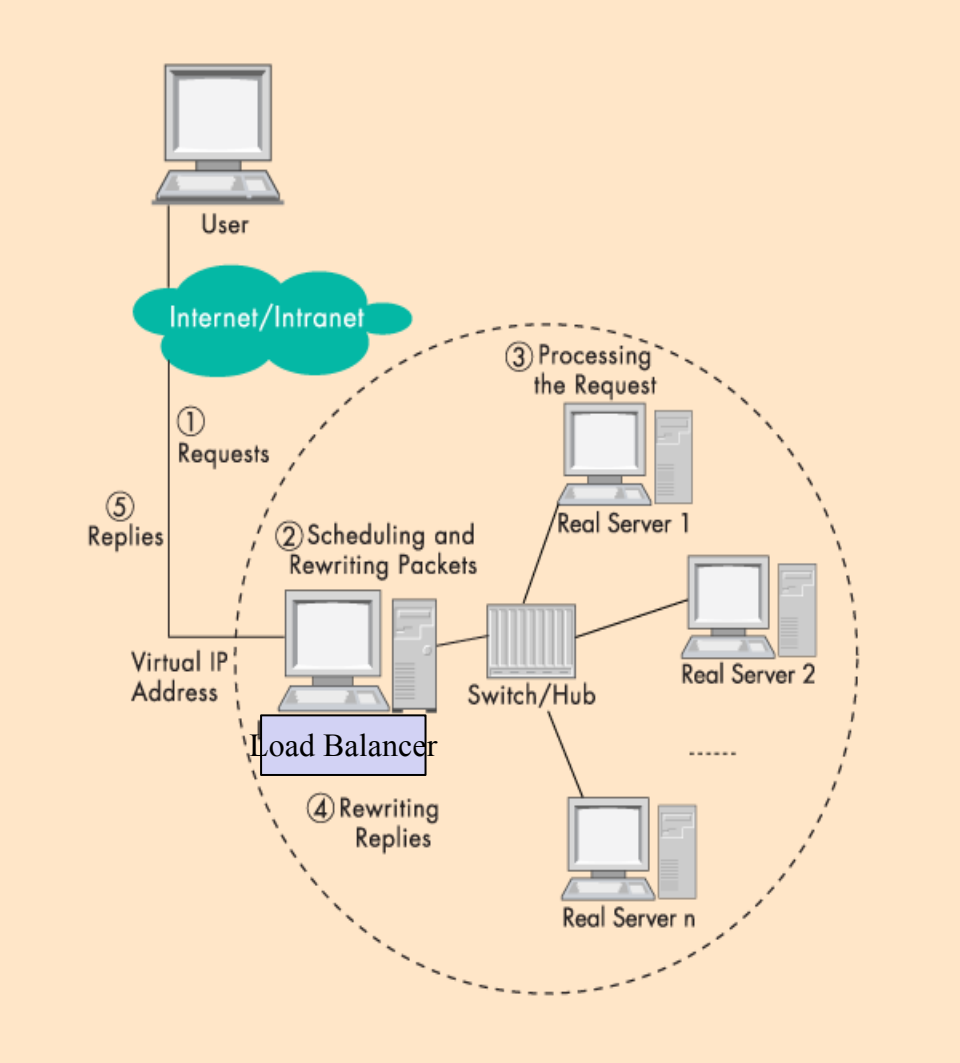
Discussion: How May You Address the Issue?

Outline

- Admin and recap
- Request routing to multiple servers
  - overview
  - DNS request routing
  - Network request routing
    - Overview of structure and issue
    - Routing direction
      - NAT

Solution 1: Network Address Translation (NAT)

- Assumption:
  - Real servers use RIPs
- Solution: NLB does rewriting/translation
- Thus, the NLB is similar to a typical NAT gateway with an additional scheduling function



Example Virtual Server via NAT

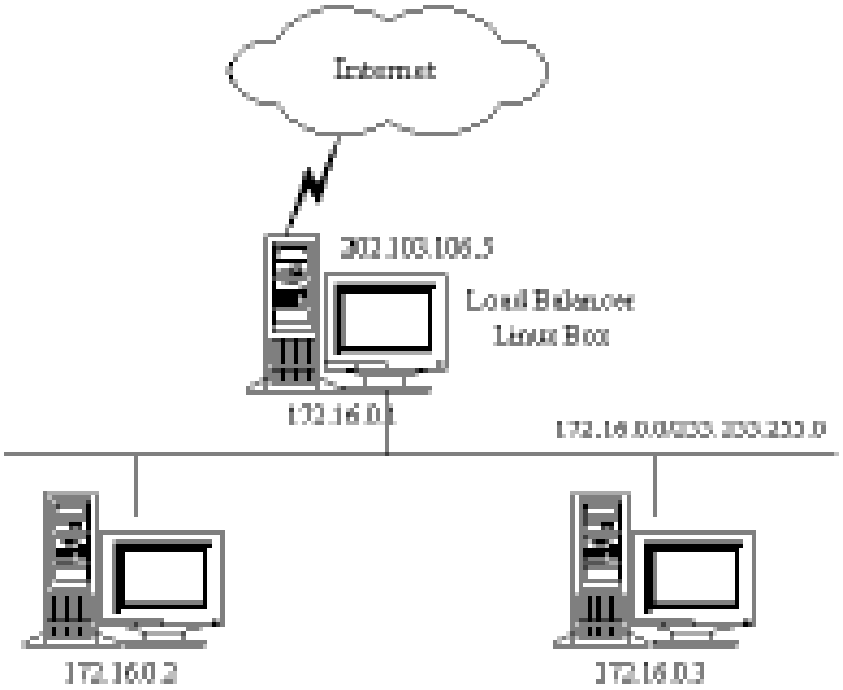
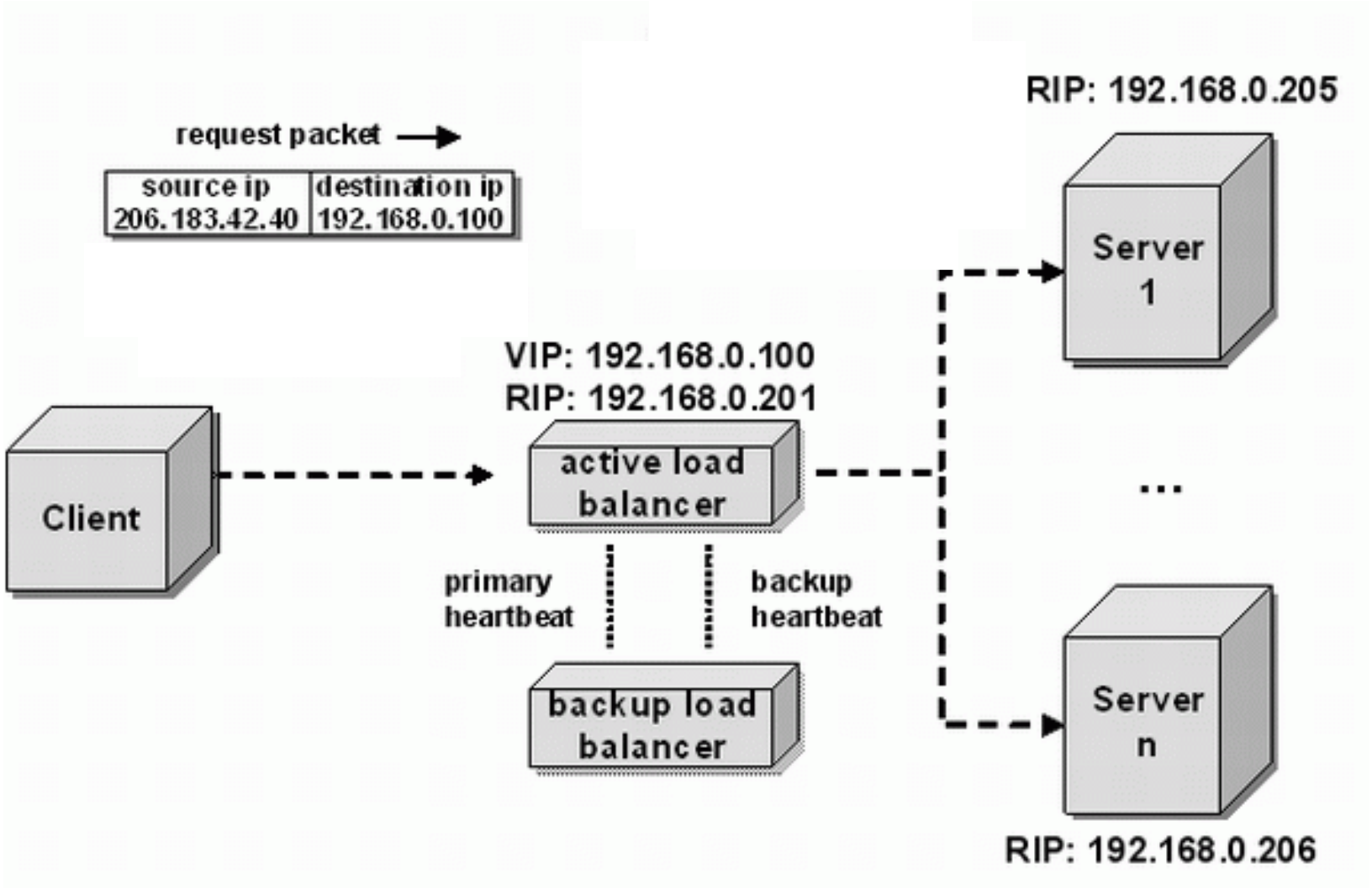


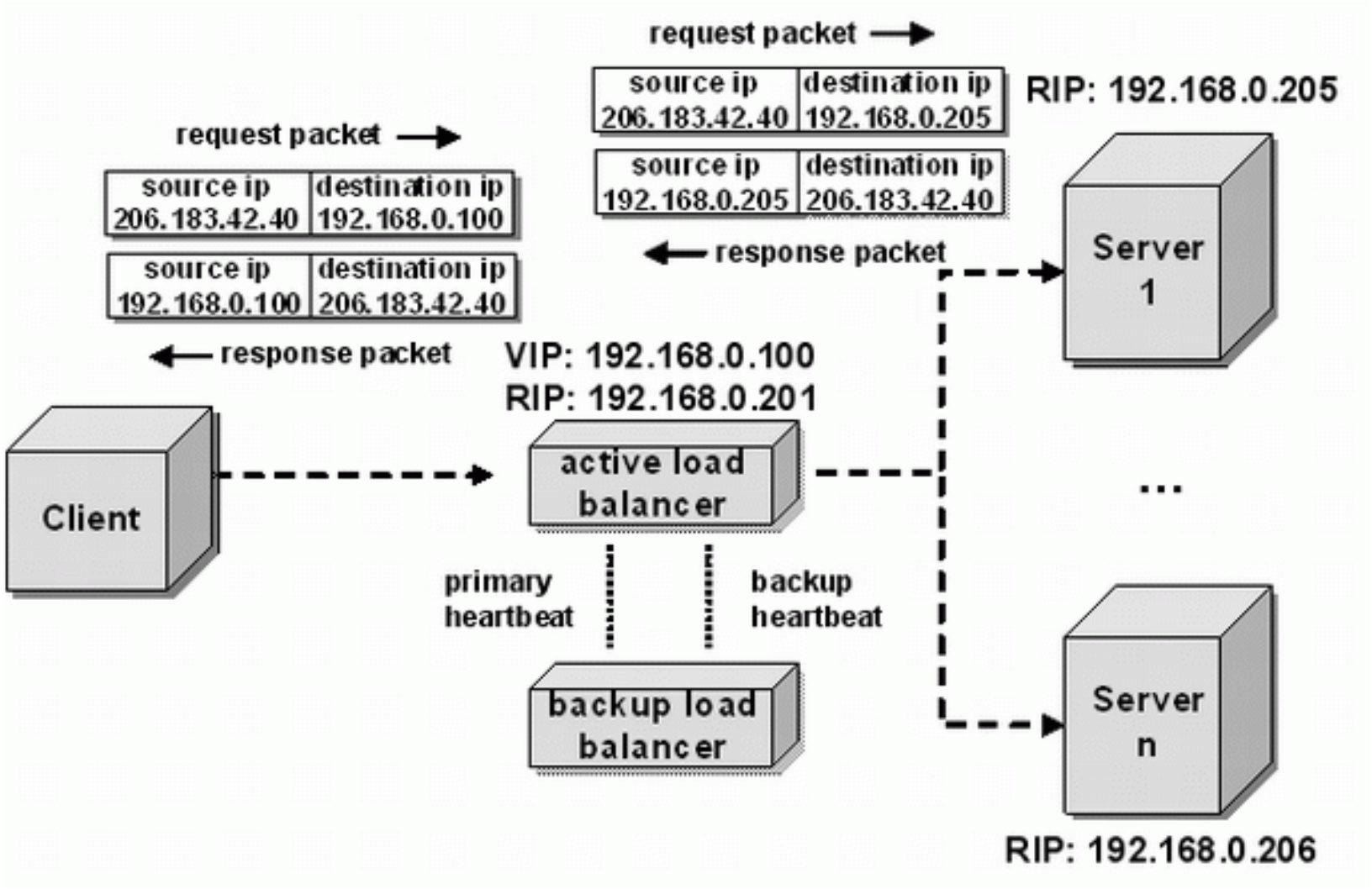
Table 1: an example of virtual server rules

Protocol	Virtual IP Address	Port	Real IP Address	Port	Weight
TCP	202.103.106.5	80	172.16.0.2	80	1
			172.16.0.3	8000	2
TCP	202.103.106.5	21	172.16.0.3	21	1

NLB/NAT Flow


request packet
source ip
206.183.42.40
destination ip
192.168.0.100
RIP: 192.168.0.205
Server 1
VIP: 192.168.0.100
RIP: 192.168.0.201
Client
active load balancer
primary heartbeat
backup heartbeat
backup load balancer
...
Server n
RIP: 192.168.0.206

NLB/NAT Flow


request packet
source ip 206.183.42.40
destination ip 192.168.0.205
RIP: 192.168.0.205
request packet
source ip 206.183.42.40
destination ip 192.168.0.100
source ip 192.168.0.205
destination ip 206.183.42.40
response packet
Server 1
source ip 192.168.0.100
destination ip 206.183.42.40
response packet
VIP: 192.168.0.100
RIP: 192.168.0.201
Client
active load balancer
...
primary heartbeat
backup heartbeat
backup load balancer
Server n
RIP: 192.168.0.206

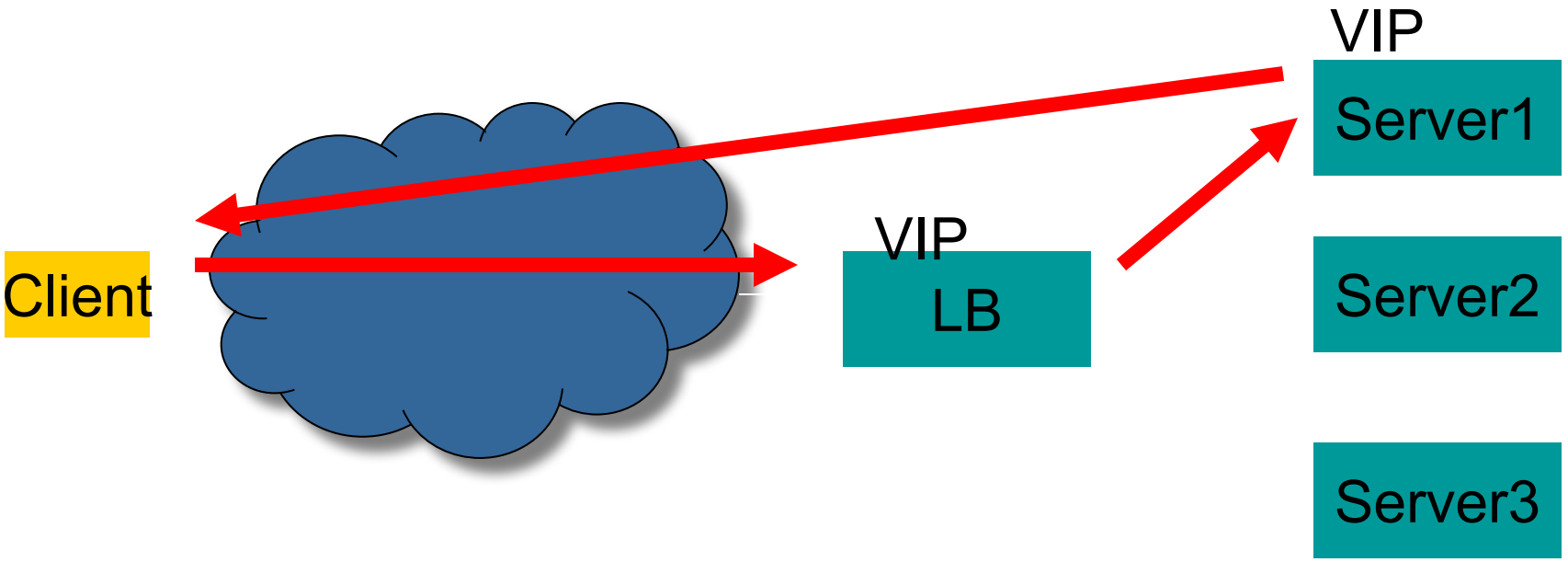
NLB/NAT Advantages and Disadvantages

- Advantages:
  - Naming abstraction: A single public IP address to be realized, transparently, by a set of real servers with private IP addresses
  - Real servers need no change and are not aware of load balancing
- Problems
  - The network load balancer must be on the critical path and hence may become the bottleneck due to load to rewrite request and response packets
    - Typically, rewriting responses has more load because there are more response packets

Outline

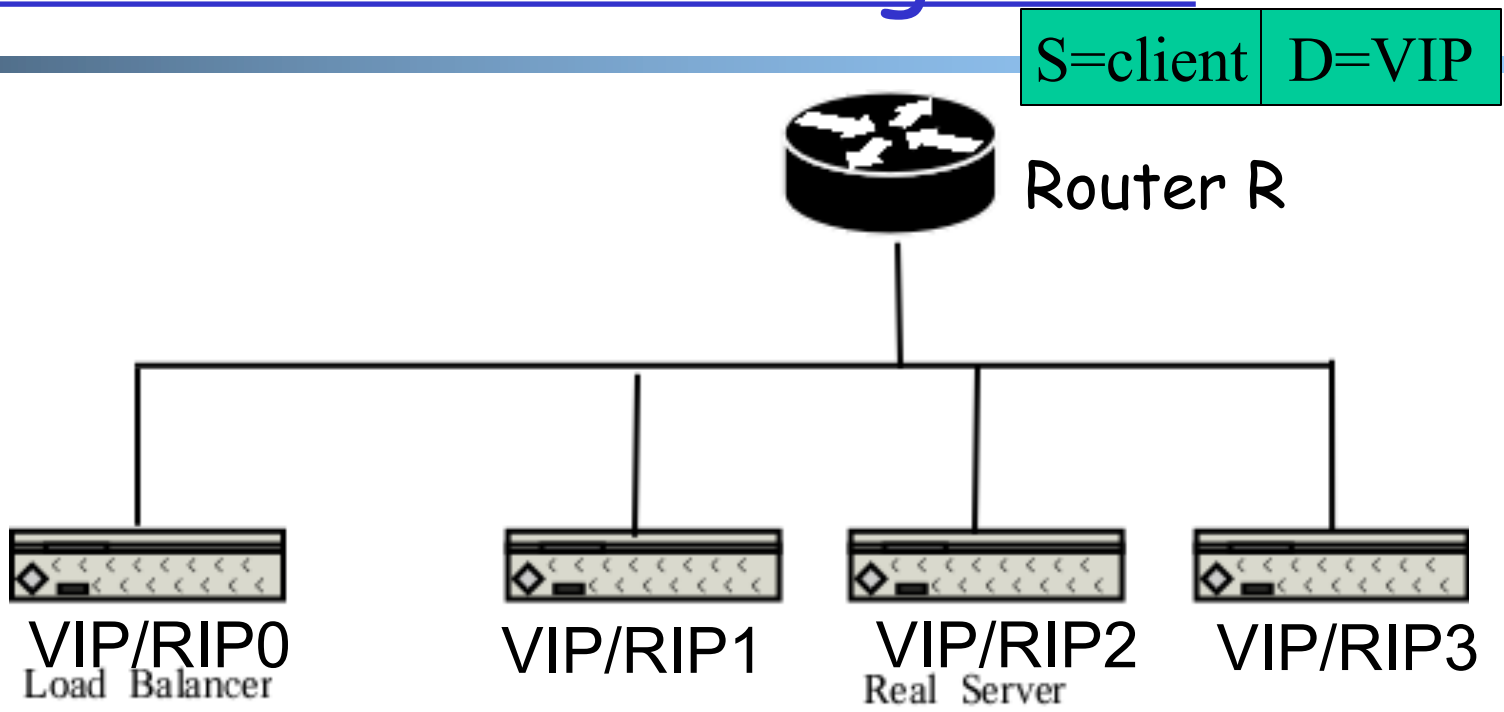
- Admin and recap
- Request routing to multiple servers
  - overview
  - DNS request routing
  - Network request routing
    - Overview of structure and issue
    - Routing direction
      - NAT
      - Direct Server Return (DSR)

NLB with Direct Server Return



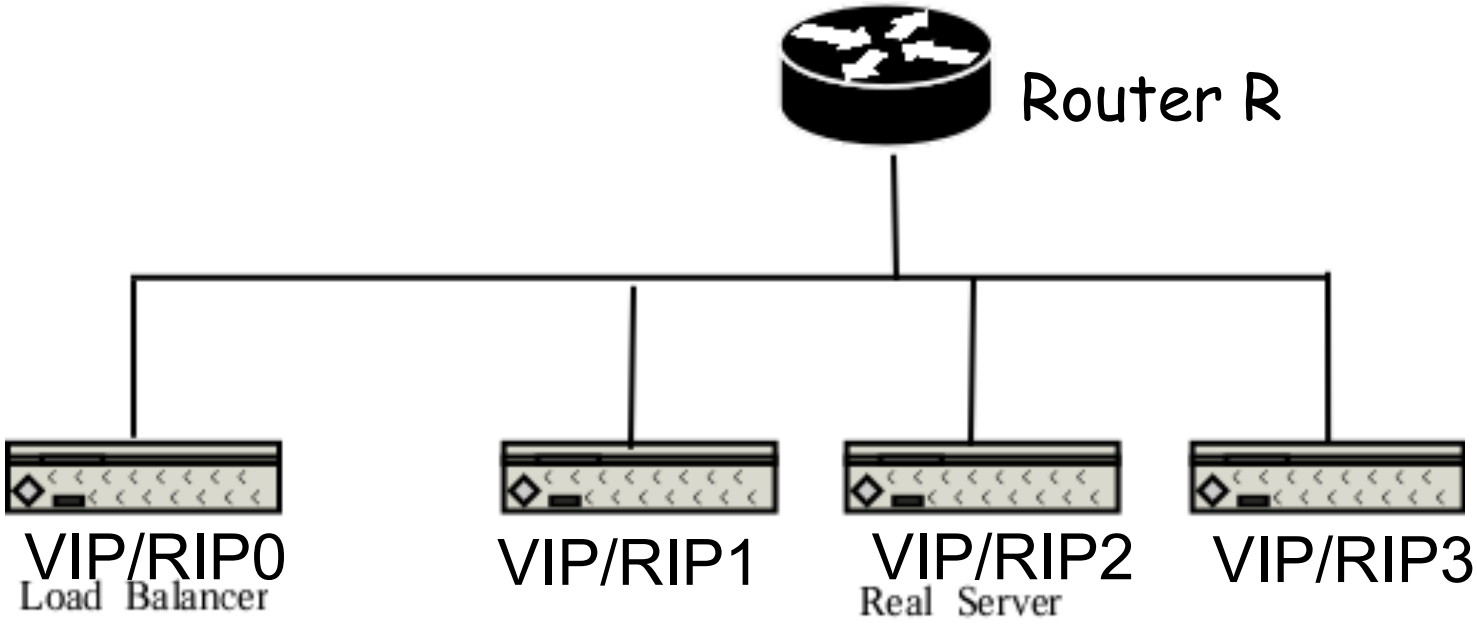
Direct server return → Each real server uses VIP as its IP address

NLB/DSR: Working Case



- Router broadcasts ARP broadcast query: who has VIP?
- ARP reply from NLB: I have VIP; my MAC is MAC_{NLB}
- Data packet from R to NLB: destination MAC = MAC_{NLB}
- How may the NLB send the request to a real server?

NLB/DSR: Problem Case



ARP race condition:

- When router R gets a packet with dest. address VIP, it broadcasts an Address Resolution Protocol (ARP) request: who has VIP?
- One of the real servers may reply before NLB

NLB via Direct DSR

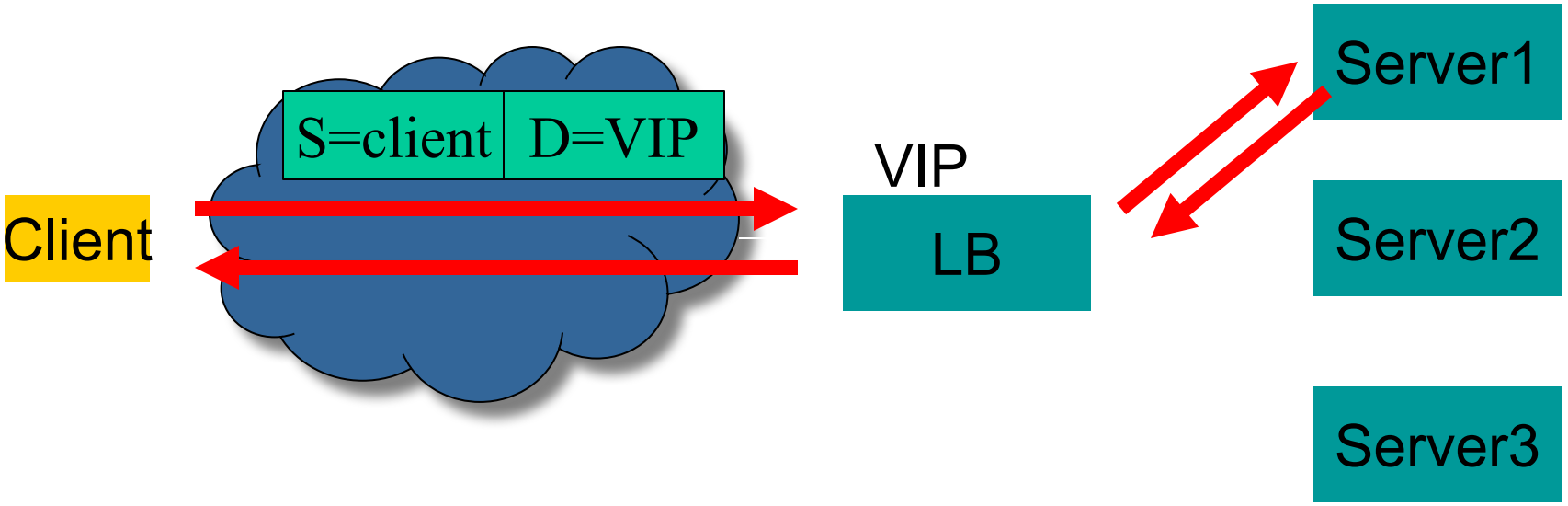
- Solution: various "hacks"
  - Configure real server with a non-ARPing, loopback alias interface with the virtual IP address, and the load balancer has an interface configured with the virtual IP address to accept incoming packets.

- The workflow of NLB/DSR is similar to that of NLB/NAT:
  - the load balancer directly routes a packet to the selected server
  - When the server receives the forwarded packet, the server determines that the packet is for the address on its loopback alias interface, processes the request, and finally returns the result directly to the user

NLB/DSR Advantages and Disadvantages

- Advantages:
  - Real servers send response packets to clients directly, avoiding NLB as bottleneck
- Disadvantages:
  - Servers must be configured specially (e.g., a non-arp alias interface for the VIP)

Discussion: Problem of Network Load Balancer Architecture So Far



A major remaining problem is that the NLB becomes a single point of failure (SPOF).

Outline

- Admin and recap
- Request routing to multiple servers
  - overview
  - DNS request routing
  - Network request routing
    - Overview of structure and issue
    - Routing direction
      - NAT
      - Direct Server Return (DSR)
    - Director reliability

Fully Distributed Directors: Microsoft NLB

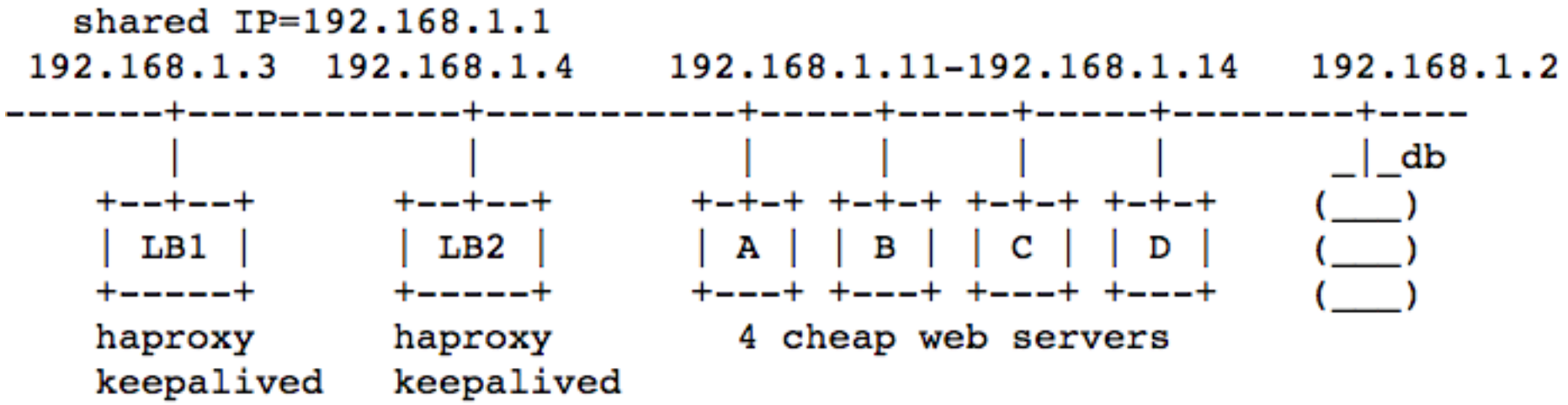
- No dedicated load balancer at all
- All servers in the cluster receive all packets
- All servers within the cluster simultaneously run a mapping algorithm to determine which server should handle the packet. Those servers not required to service the packet simply discard it.
  - Mapping (ranking) algorithm: computing the "winning" server according to host priorities, multicast or unicast mode, port rules, affinity, load percentage distribution, client IP address, client port number, other internal load information

http://technet.microsoft.com/en-us/library/cc739506%28WS.10%29.aspx

Discussion

- Advantages and issues of fully distributed NLB

Active/Passive Request Routers: HAProxy using VRRP [RFC3768]



Configuration on LB1/LB2

```
listen webfarm 192.168.1.1:80
   mode http
   balance roundrobin
   cookie JSESSIONID prefix
   option httpclose
   option forwardfor
   option httpchk HEAD /index.html HTTP/1.0
   server webA 192.168.1.11:80 cookie A check
   server webB 192.168.1.12:80 cookie B check
   server webC 192.168.1.13:80 cookie C check
   server webD 192.168.1.14:80 cookie D check
```

Configuration keepalived LB1/LB2

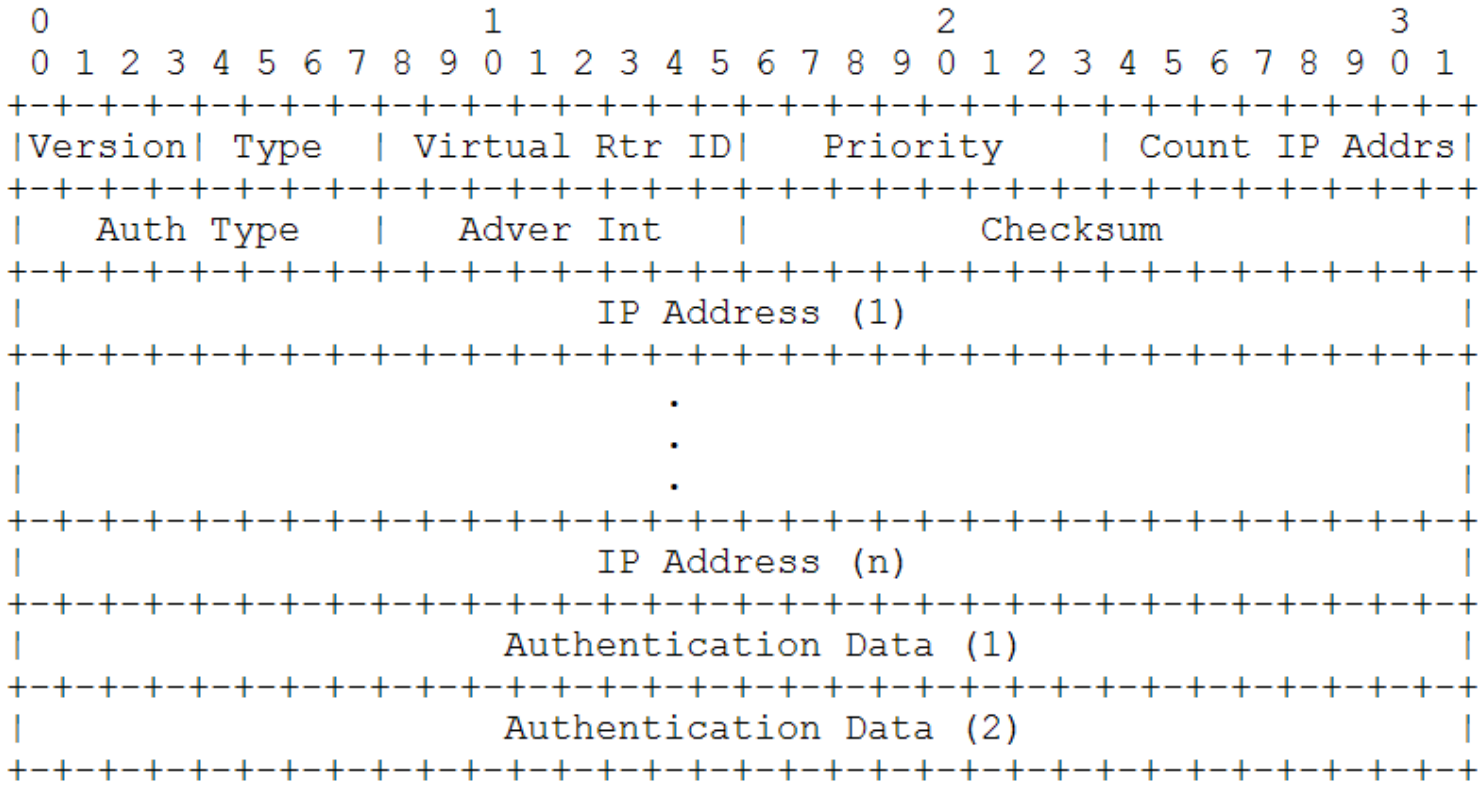
```
vrrp_script chk_haproxy {           # Requires keepalived-1.1.13
    script "killall -0 haproxy"     # cheaper than pidof
    interval 2                      # check every 2 seconds
    weight 2                        # add 2 points of prio if OK
}

vrrp_instance VI_1 {
    interface eth0
    state MASTER
    virtual_router_id 51
    priority 101                    # 101 on master, 100 on backup
    virtual_ipaddress {
        192.168.1.1
    }
    track_script {
        chk_haproxy
    }
}
```

https://www.haproxy.org/download/1.2/doc/architecture.txt

Virtual Router Redundancy Protocol: Basic Ideas

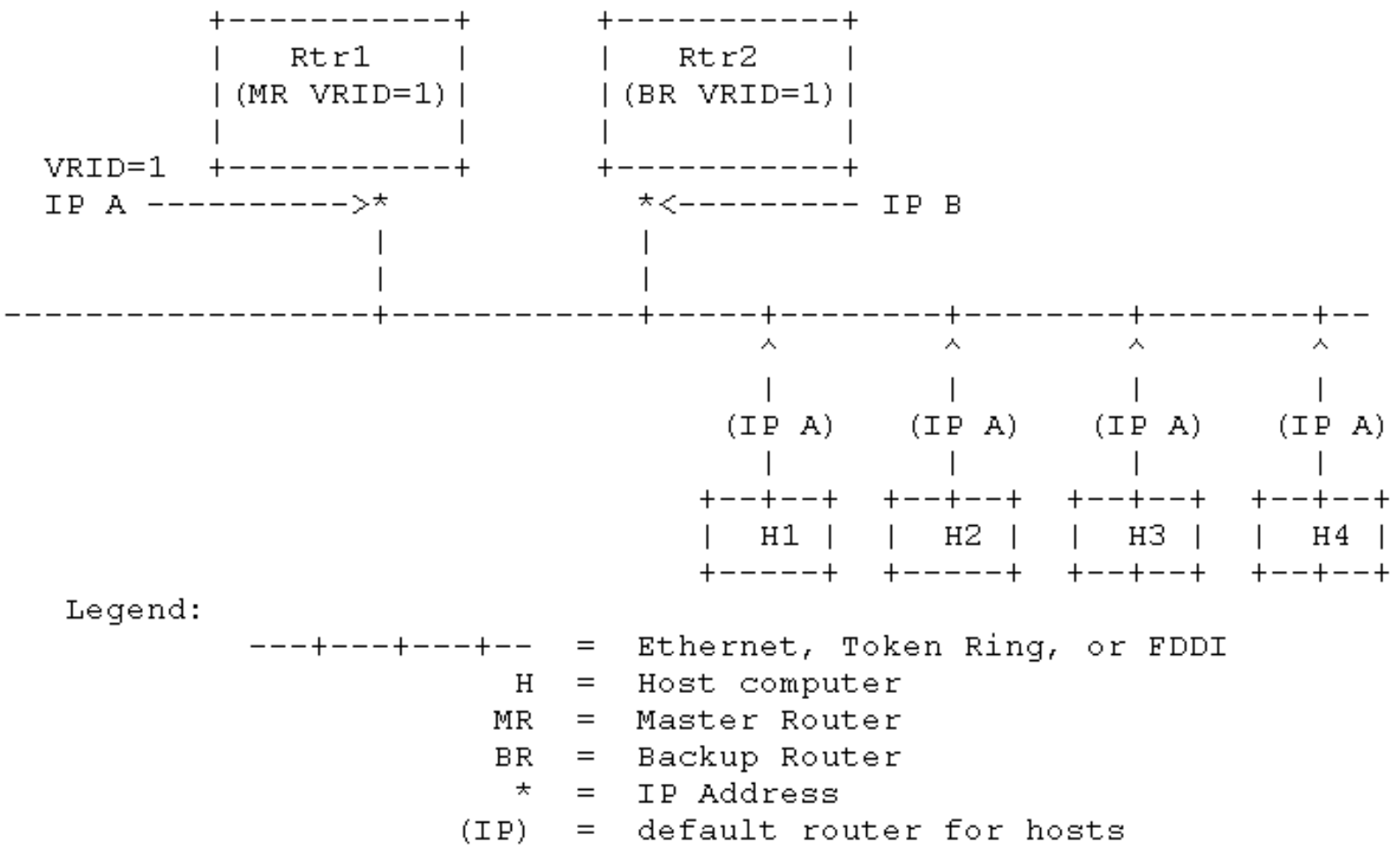
- Virtual router
  - Specified by a virtual router identifier (VRID) and a set of associated IP addresses
  - Each virtual router ID has a corresponding (virtual) MAC: 00-00-5E-00-01-[VRID]
  - Leader election among the physical routers to select a single master, who
    - Owns the given IPs of the virtual router
    - Receives and forwards packets for the IPes

Virtual Router Redundancy Protocol: Protocol and Msgs



- Periodical, multicast announcements to IP address 224.0.0.18 w/ IP protocol number 12

VRRP Sample Configuration



Discussion

- Advantages and issues of active/passive request routing