CSC7970 – NEXT-GENERATION NETWORKING

CONTENT DELIVERY NETWORKS

Instructor: Susmit Shannigrahi
sshannigrahi@tntech.edu

CDNs

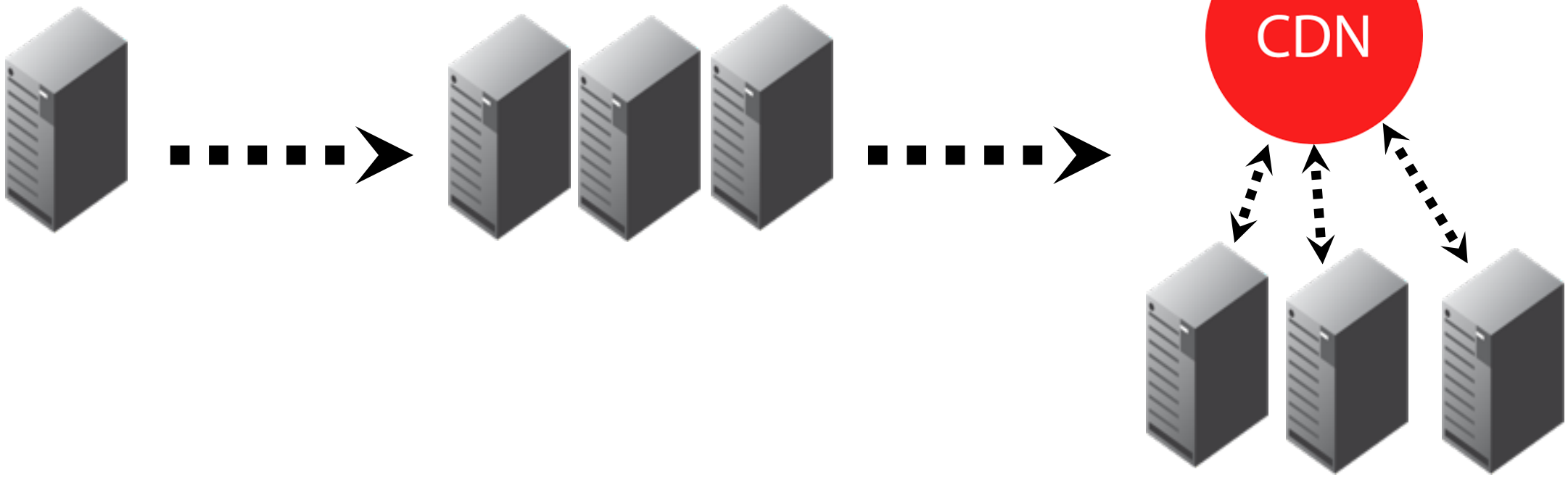
- Content is the primary focus of the Internet
  - What was the Internet designed for?
  - Content distribution requires workarounds
- Main idea
  - Move content near the user using geographically distributed servers
  - Cache popular content in a application overlay
- ***Goal: S*calable** content distribution
  - Example: Stranger things

The CDN model

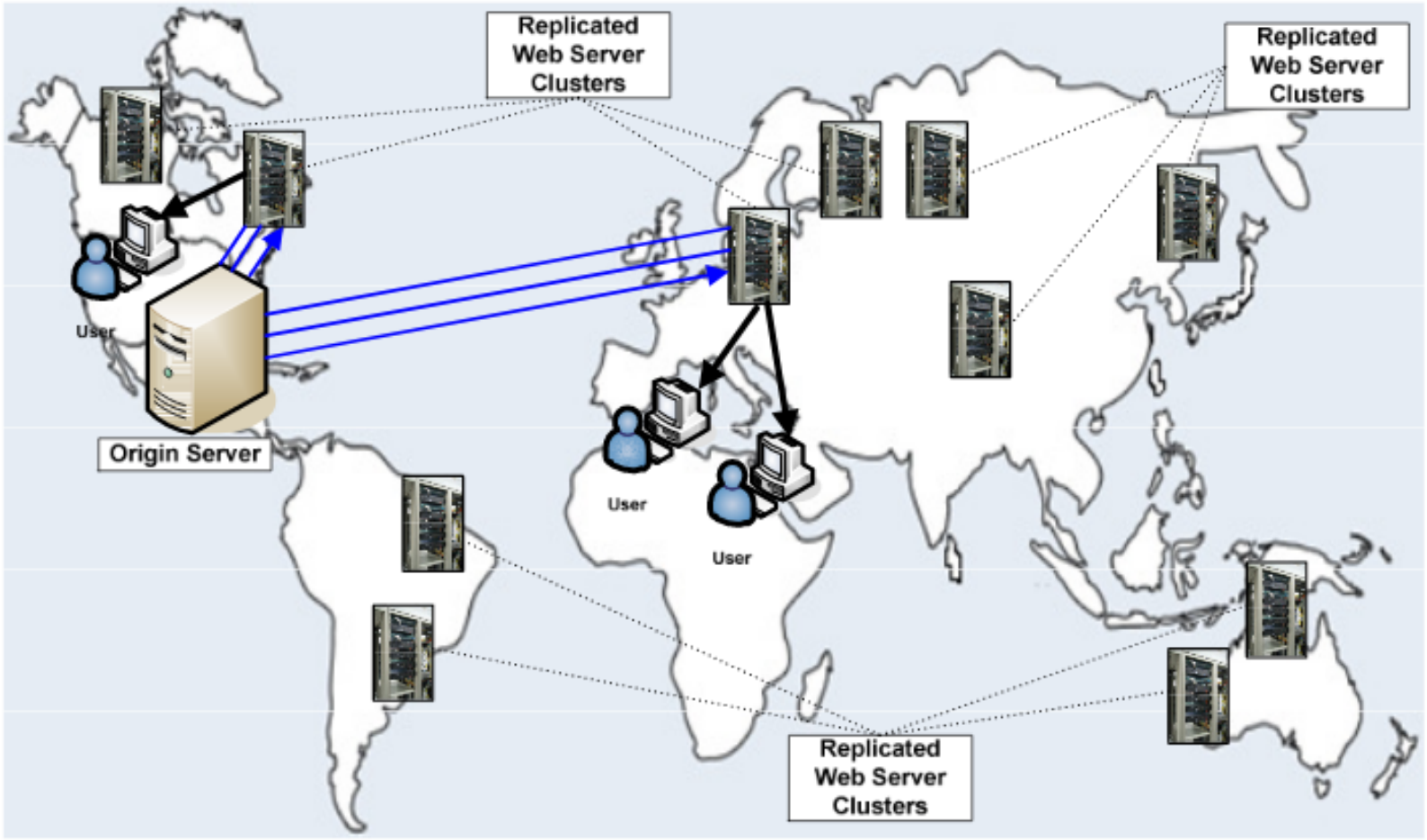
Single server model

Multi-server model
(replicated content)

CDN model



CDN Entities



- Origin Server
- Replica Servers
- Clients
- Point of Presence (POP)
- Caches
- DNS

http://www.cloudbus.org/reports/CDN-Taxonomy.pdf

Who uses CDNs?

- Everyone
  - But mainly geared towards content distribution
- Why would a content owner use them?
  - TCP/IP Internet architecture does scale for content distribution
  - Load and performance challenges
  - Improved response times

How important can “latency” be?

- Affects user experience
  - Users lose attention if a page takes more than a few hundred ms to load
  - 2 Seconds is an eternity

- Affects corporate revenue. Examples:
  - *Amazon*: revenue increase of 1% for every 100ms of reduction in page response time
  - Google experienced a 20% decrease in ad revenue with a half-second increase in page load time.
  - *Shopzilla*: revenue increase of 12% by reducing page response time from 6 seconds to 1.2 seconds

https://blog.uptrends.com/web-performance/the-psychology-of-web-performance/

Who provides CDN services?

- Also everyone:
  - Akamai started it, Amazon, Google, MS, Verizon,

- Big companies have their own CDNs
  - Why?
  - Expensive
  - If a provider fails, their data becomes unavailable
  - Vendor lock-in
  - Costs money to migrate

Percent of content coming from CDNs

- Any guess/idea?

- *Conservative estimation:* half of the bits of Internet traffic
- *Probably closer to reality*: 80-90% of the bits of Internet traffic

“Reverse-engineering” CDNs

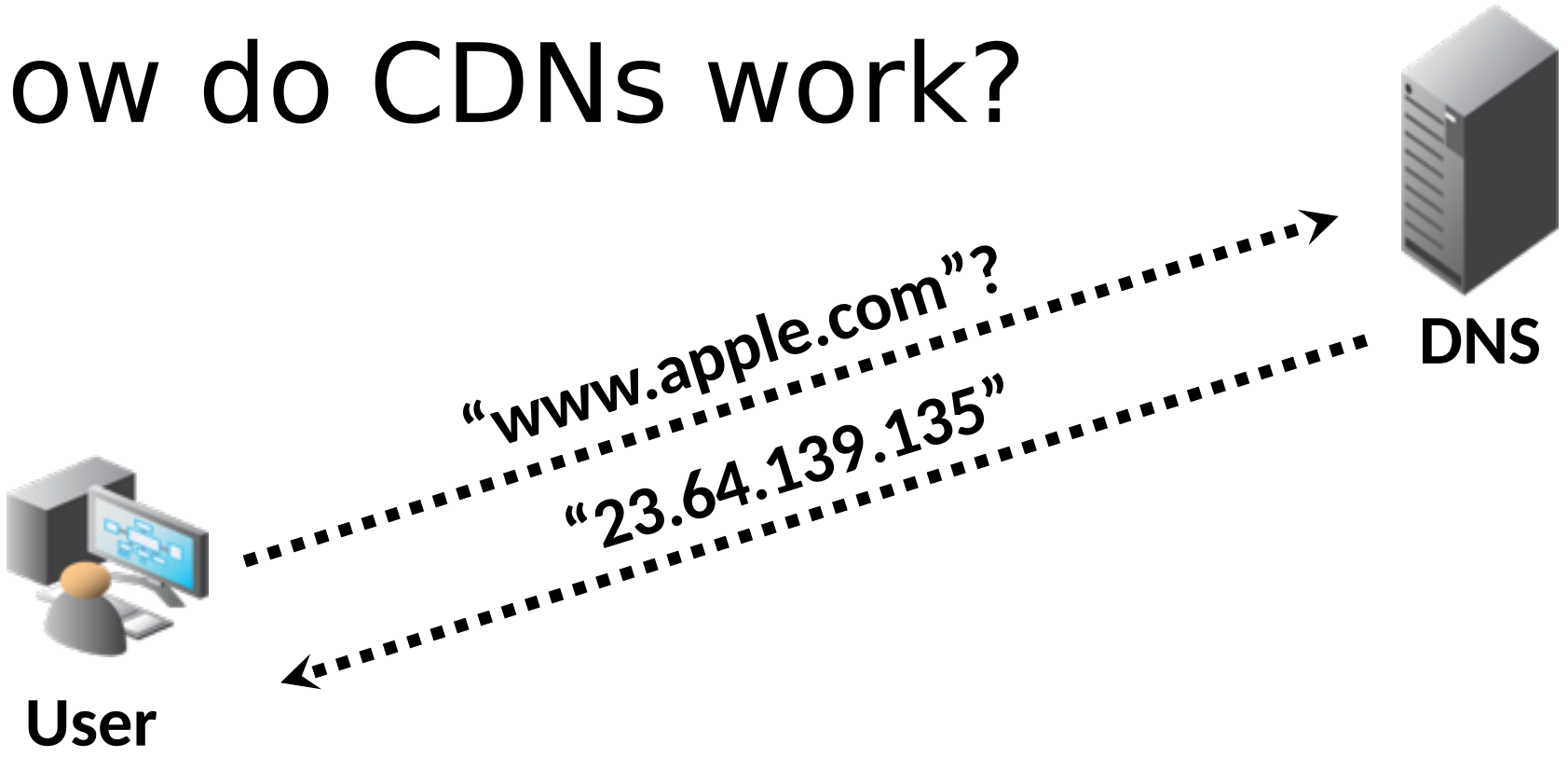
```
Spyros$ dig www.apple.com

; <<>> DiG 9.10.6 <<>> www.apple.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57705
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.apple.com.                 IN      A

;; ANSWER SECTION:
www.apple.com.          57      IN      CNAME   www.apple.com.edgekey.net.
www.apple.com.edgekey.net. 16510 IN     CNAME   www.apple.com.edgekey.net.globalredir.akadns.net.
www.apple.com.edgekey.net.globalredir.akadns.net. 1975 IN CNAME  e6858.dsce9.akamaiedge.net.
e6858.dsce9.akamaiedge.net. 14  IN      A       23.64.139.135
```

How do CDNs work?

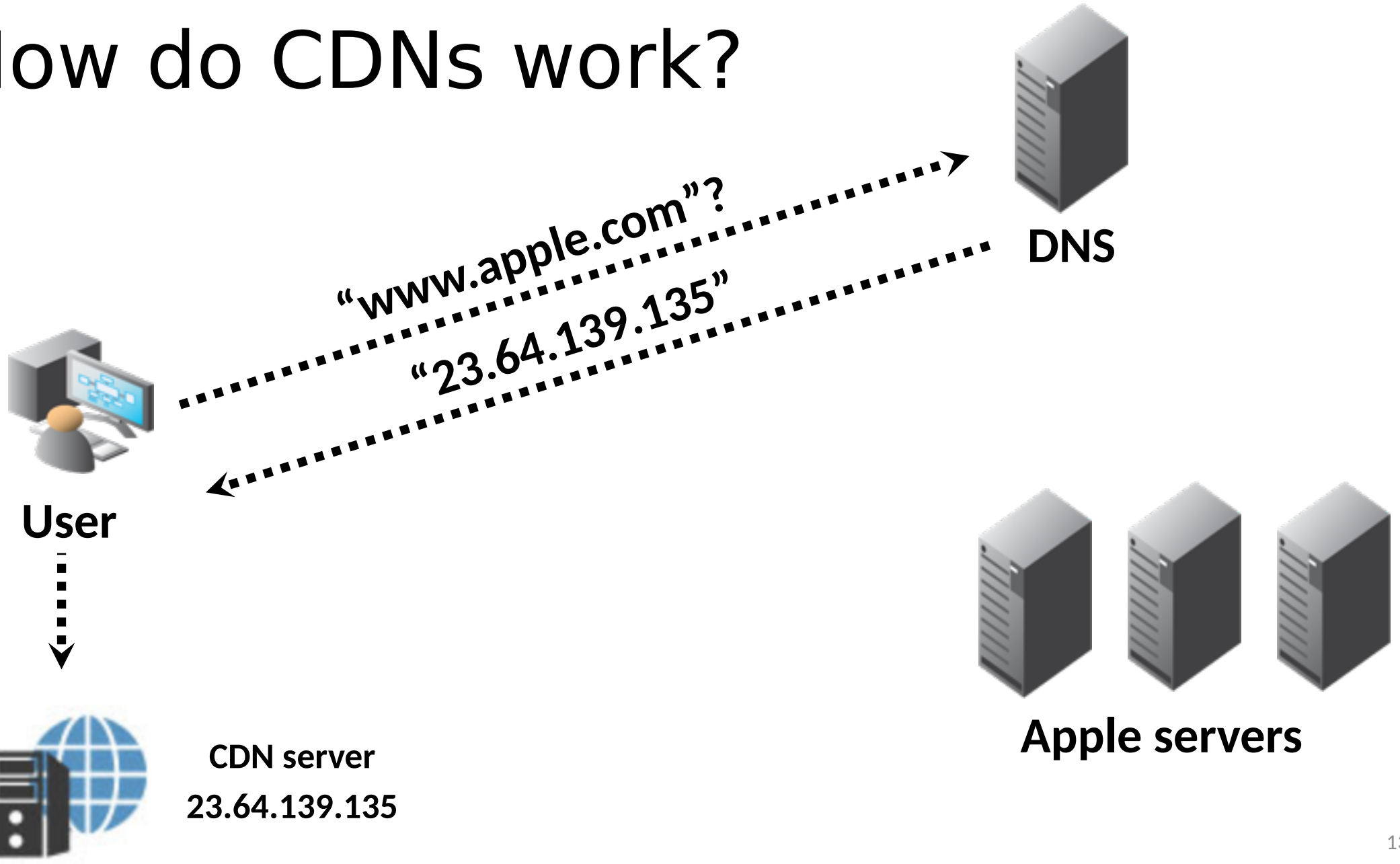
User

How do CDNs work?

How do CDNs work?



How do CDNs work?



Discovering a CDN server

- DNS redirection
  - DNS returns IP address of CDN server(s) to clients
- Advantages
  - Any ideas?
  - DNS already has scalable infrastructure
  - URLs need no change
- Limitations
  - Who controls the user content?
  - DNS typically returns a single IP address
  - This IP address might be unicast

How to get around the DNS limitations?

- IP anycast
  - Deploy multiple CDN servers with same IP address
  - All these servers have the requested content
  - Content requests reach the closest CDN server to the user

Anycast example

DNS

“www.apple.com”?

“23.64.139.135”

User

CDN server #2

23.64.139.135

CDN server #1

23.64.139.135

CDN server #3

23.64.139.135

Anycast example

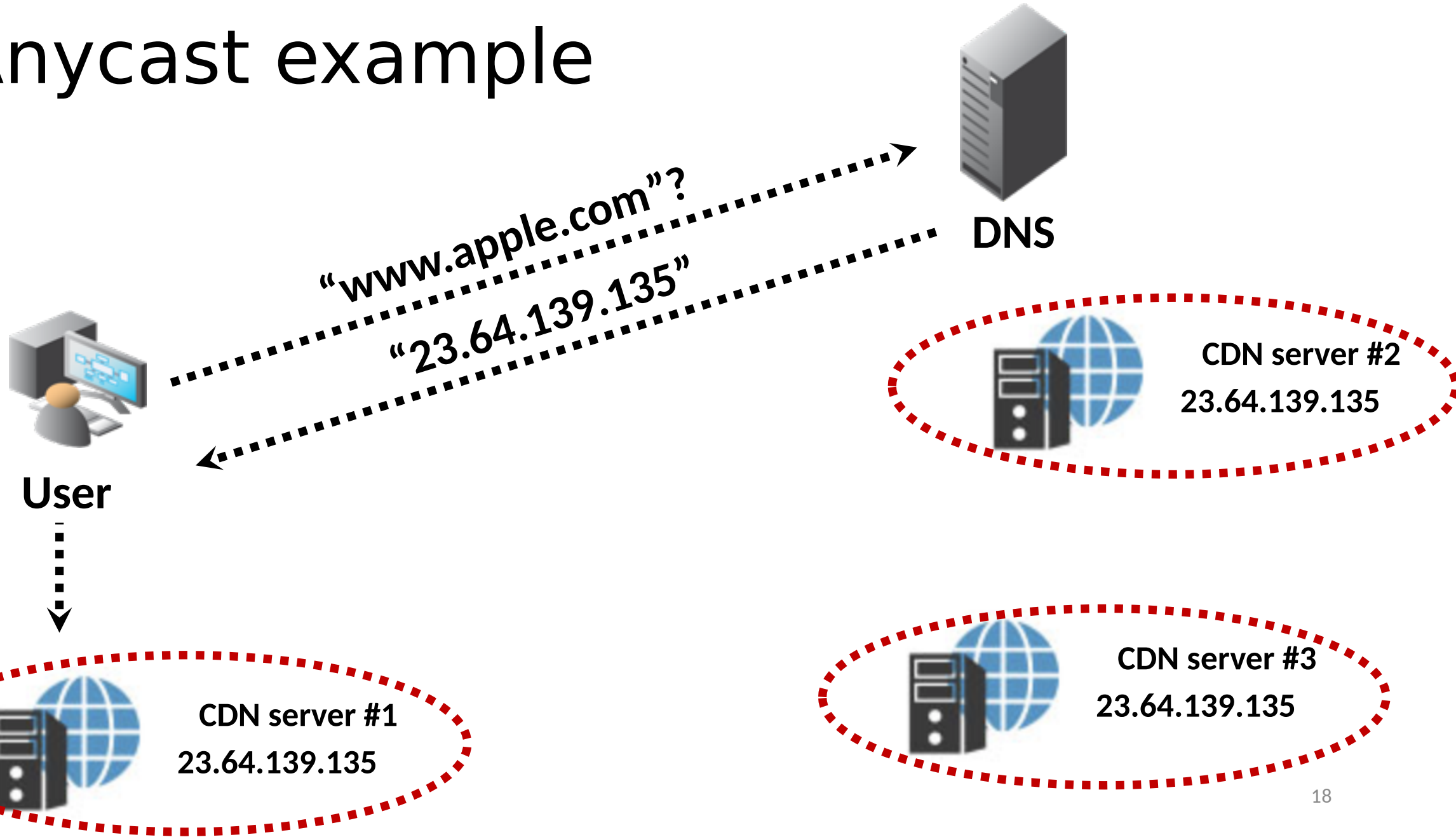
DNS

“www.apple.com”?

“23.64.139.135”

User

Anycast example



Further optimizing on performance

- Multiple CDN servers in same data-center having the requested content
  - How to pick the ”best” one?

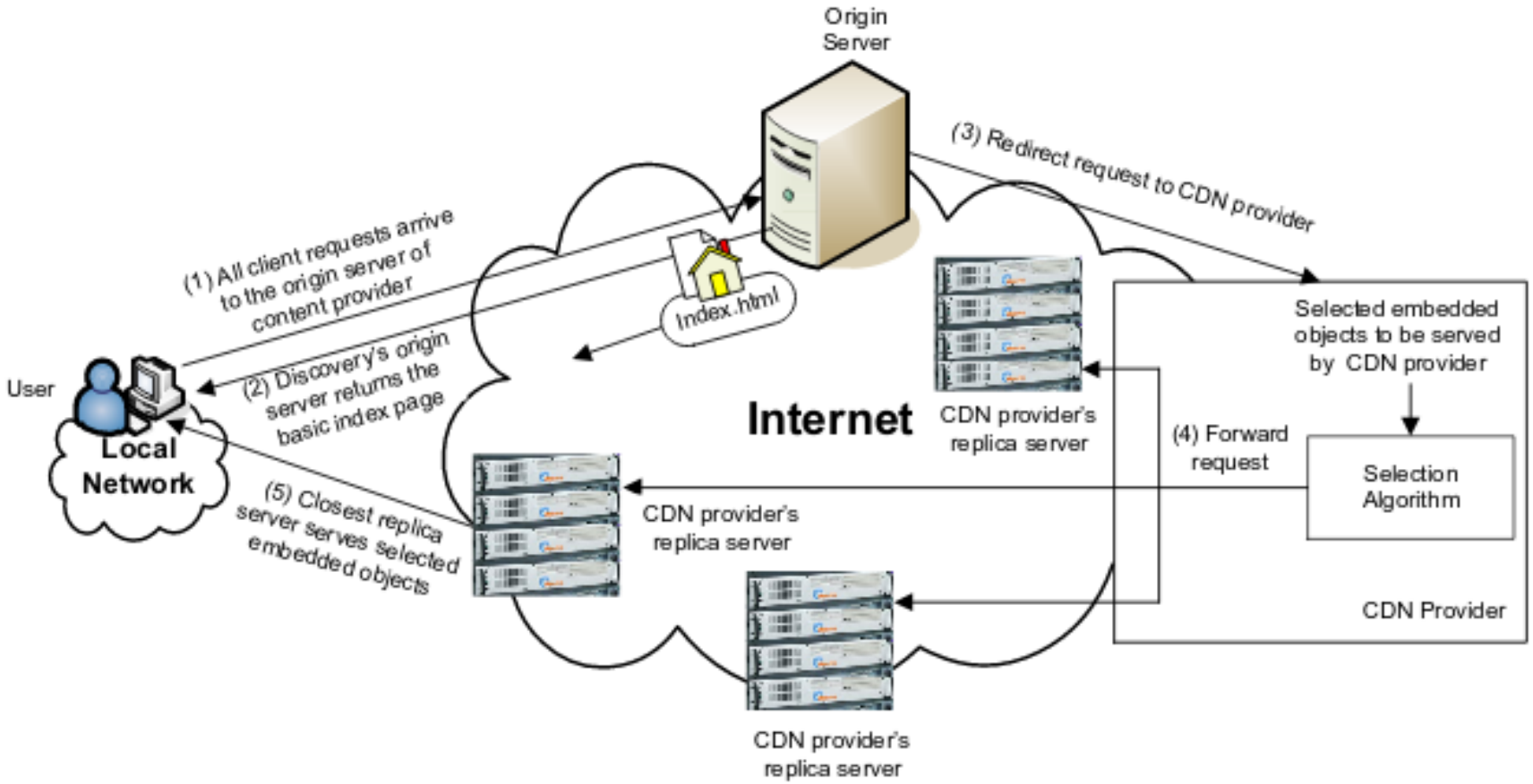
- It depends on how we define “the best one”
  - Load, network conditions, service level agreements
  - Typically solved through load balancers, network monitoring, and configuration

Request Routing

- Multiple CDN servers in same data-center having the requested content
  - How to pick the ”best” one? ← Request routing

- It depends on how we define “the best one”
  - Load, network conditions, service level agreements
  - Typically solved through load balancers, network monitoring, and configuration

Request Routing



http://www.cloudbus.org/reports/CDN-Taxonomy.pdf

How CDNs handle dynamic content?

- What to do with dynamic content?
  - Examples: webpages with ads or websites that require passwords
  - Break dynamic pages into parts that can be cached and parts that cannot
  - CDN server fetches non-cacheable parts (e.g., response to a password) from the original content server and cacheable parts from its own cache
  - CDN server puts non-cacheable and cacheable parts together, assembles the website, and sends it back to the user

- How effective could that be?
  - According to Akamai, this reduces bandwidth requirements for dynamic content by 95-99%

Security/Privacy

- Impact on security/privacy?
  - Users and content providers need to trust CDNs!
- Who has control over user content?
  - For sure, not the user!
- What happens if the CDN gets compromised?
  - Content poisoning!

Encrypted content

- Why not using end-to-end encryption to solve all the security/privacy problems?
  - CDNs need to have access to unencrypted content to optimize content delivery!

- *User think they establish an encrypted connection to the content server!*
  - ***Typically NOT true!***
  - CDNs usually have access to encryption/decryption keys!
  - CDNs fetch content from content servers --> store content (typically unencrypted) --> encrypt requested content and provide it to users

Mitigate DDoS attacks through CDNs

- Lately, DDoS attacks are against applications/services (e.g., Facebook) in addition to network services (e.g., DNS)
  - Attack a large distributed network of caches harder than attacking a few content servers directly!
  - Attacks against CDNs (big guys) instead of content owners (smaller guys)
  - CDNs monitor traffic -> detect anomalies and attacks
  - Drop or reroute malicious traffic almost in real-time

Conclusion

- CDNs are ubiquitous now
- CDNs are still under active development
- Many free CDNs you can use today
  - Cloudflare, netli and many others

Akamai Paper

- Main takeaways
  - CDNs started for solving the flash crowd problem
  - Multi-homing, clusters, mirroring do not scale
  - Define network functions – nearest, available, and likely
  - DNS + Defined Network Functions for choosing servers
  - Dynamically assemble content for delivery
  - For streaming content – get data from origin server and send to the edge, users get content from the edge
- Massively replicate content and services
- Versioned objects for content delivery
  - Close communication with content providers (trust)

Netflix Paper

- Main takeaways
  - How to measure distributed infrastructure (again)
  - Netflix owns its CDN – cost
  - Distribute a plugin and convince people to use it, look at the requests and traffic
  - Server names are location specific and has a cache numner
    - ipv6_1-lagg0-c002.1.lhr005.bt.isp.nflxvideo.net
    - Well defined and hierarchical
  - Massive scale
    - ~4300 servers in the US alone! (Mostly at IXPs)
    - No deployment in major ISPs – they want Netflix to pay for traffic
    - In Brazil – the opposite – mostly at ISPs – why?
  - What about complexity?