Traffic Engineering for CDNs

Matt Jansen
Akamai Technologies

BDNOG3, Dhaka, May 19th 2015

The Akamai Intelligent Platform

The world's largest on-demand, distributed computing platform delivers all forms of web content and applications

The Akamai Intelligent Platform:

170,000+ Servers	2,000+ Locations	1,300+ Networks	700+ Cities	102+ Countries

Typical daily traffic:

- More than **2 trillion** requests served
- Delivering over **25 Terabits/second**
- **15-30%** of all daily web traffic

How CDNs Work

When content is requested from CDNs, the user is directed to the optimal server to serve this user

There's 2 common ways to do that:

- anycast: the content is served from the location the request is received (easy to build, requires symmetric routing to work well)
- DNS based: the CDN decides where to best serve the content from based on the resolver it receives the request from, and replies with the optimal server

How anycast based CDNs Work

The CDN announces the same IPs in multiple places (anycast)

The routing protocol will pick the ‘best’ path to the (topologically) closest location.

The CDN replies from where it receives the request

How DNS based CDNs Work

Users querying a DNS-based CDNs will be returned different A (and AAAA) records for the same hostname depending on the resolver the request comes from

This is called “mapping”

The better the mapping, the better the CDN

DNS based Mapping Example

Example of Akamai mapping

- Notice the different A records for different locations:

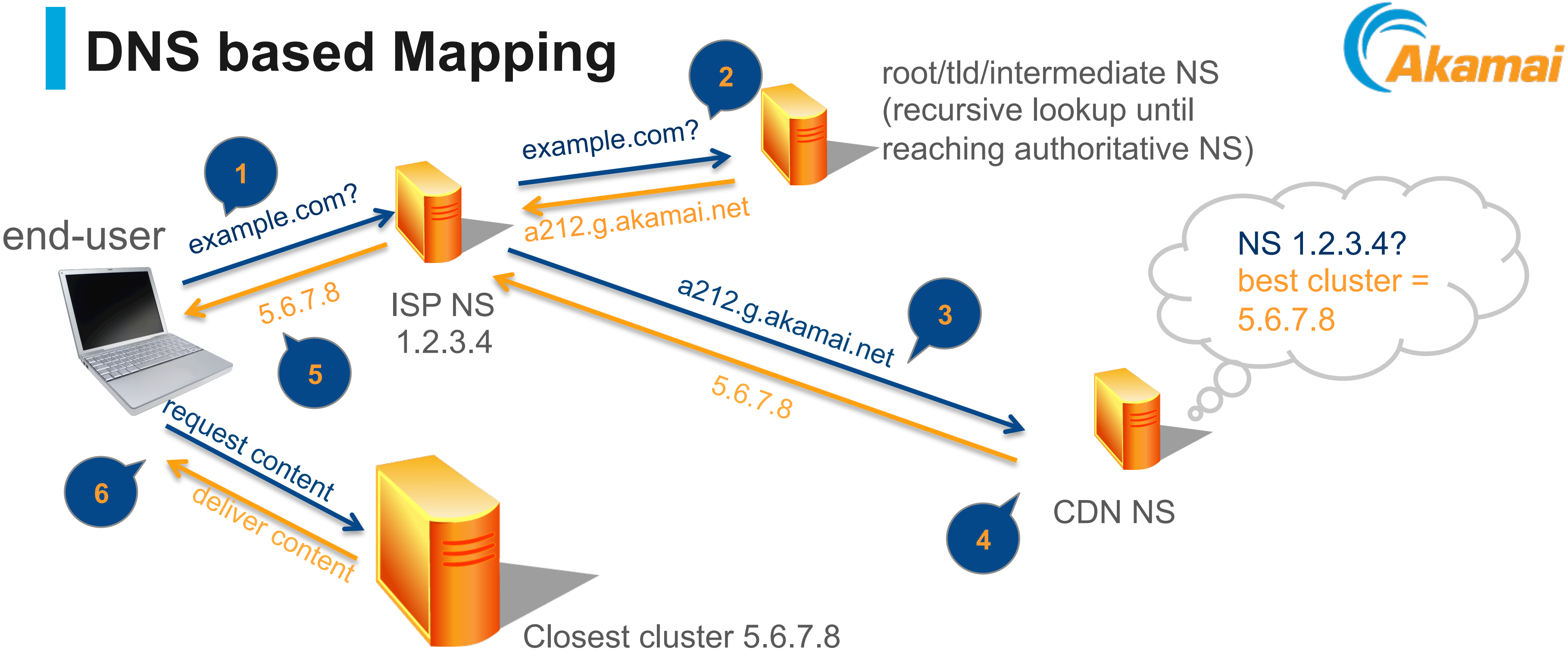
```
[NYC]% host www.symantec.com
www.symantec.com   CNAME  e5211.b.akamaiedge.net.
e5211.b.akamaiedge.net.  A      207.40.194.46
e5211.b.akamaiedge.net.  A      207.40.194.49

[Boston]% host www.symantec.com
www.symantec.com   CNAME  e5211.b.akamaiedge.net.
e5211.b.akamaiedge.net.  A      81.23.243.152
e5211.b.akamaiedge.net.  A      81.23.243.145
```

How DNS based CDNs Work

CDNs use multiple criteria to choose the optimal server

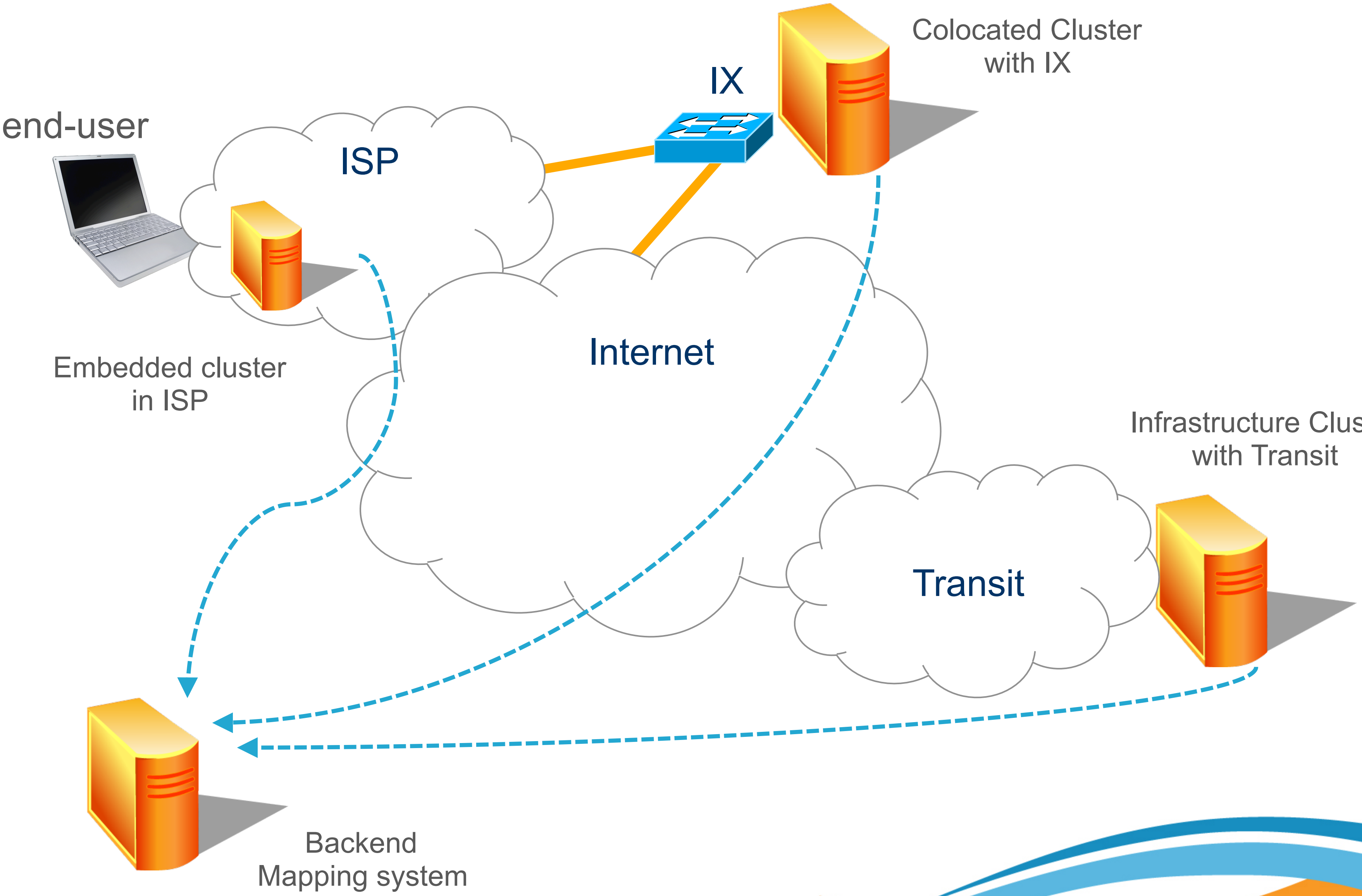
- These include standard network metrics:
  - Latency
  - Throughput
  - Packet loss
- as well as internal ones such as:
  - CPU load on the server
  - HD space
  - network utilization

DNS based Mapping



1) end-user requests www.example.com from ISP NS
2) ISP NS recursively (multiple iterations) looks up www.example.com being referred to authoritative CDN NS (by CNAME)
3) ISP NS asks authoritative CDN NS
4) CDN NS looks up IP of requestor (ISP NS) and replies with IP of optimal cluster to serve content (closest cluster for that ISP)
5) ISP NS replies to end-user who
6) requests content from local Cluster

Types of CDN Deployments (simplified)



Typical DNS based Topology

DNS based CDNs don’t necessarily have a backbone

Those clusters can be standalone ‘Islands’ with no connectivity between them!

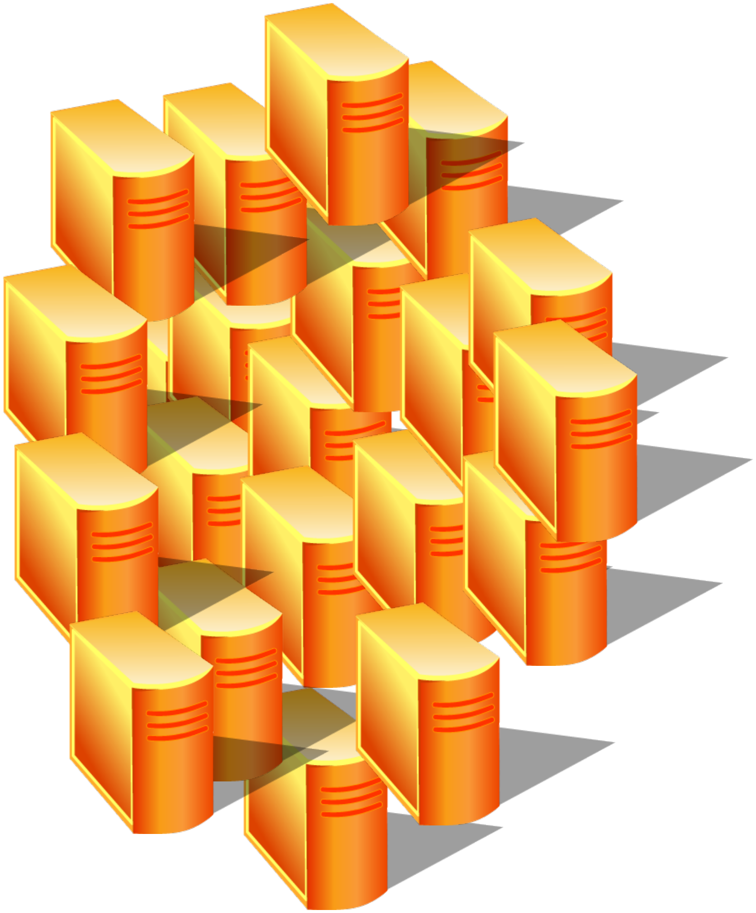
DNS based CDNs usually don’t announce large blocks of address space as this will only be the local servers

- it is not uncommon to see a single /24 from Akamai at an IX
- you will receive a different set of prefixes on each peering

This does not mean you will not see a lot of traffic

- how many servers do you need to serve 10g nowadays?

Why don’t I get all CDN Traffic locally?



- no single cluster can accommodate all content
- caches get more efficient with with size
- some content requires specialized servers only present in Infrastructure clusters
- some content is only present in specific geographies
- CDNs might prefer on-net cluster over peering

Why CDNs peer

Performance

- getting as close as possible to the end-user (removing intermediate ASNs) decreasing latency/increasing troughput

Burstability

- for large event peaks direct connectivity to multiple networks allows for higher burstability than a single connection to a transit provider

Redundancy

- Serve as overflow and backup for embedded on-net clusters

Scale

- Large deployment at an IX can serve traffic that does not ‘fit’ into smaller embedded clusters

Why ISPs peer with CDNs

Performance

- ISP’s end-users benefit from direct connectivity

Competitive Advantages

- improving performance over competitors
- additional revenue from downstreams

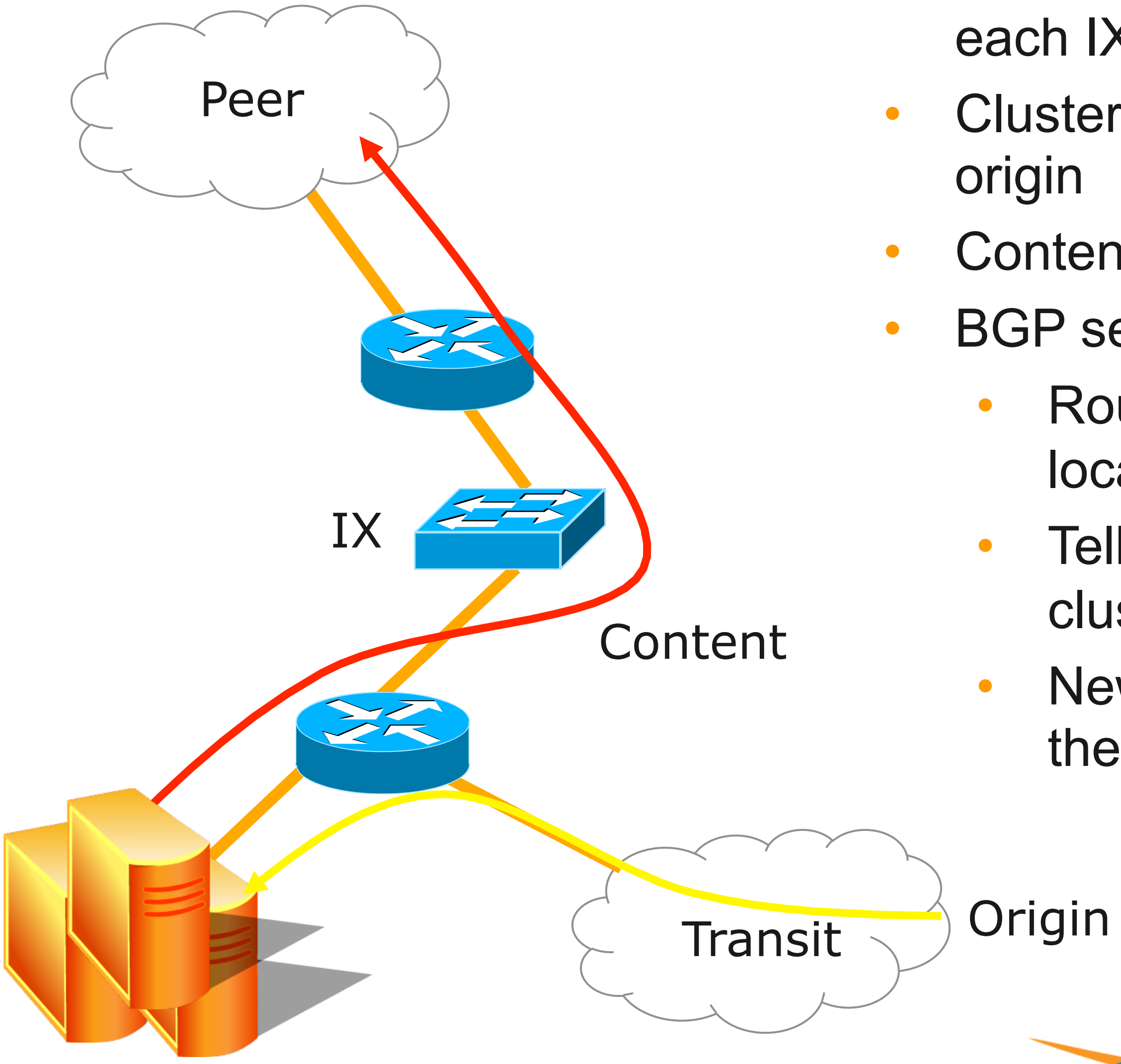
Cost Reduction

- Save on transit bill and potential backbone costs

Redundancy

- Serve as overflow and backup for embedded on-net clusters

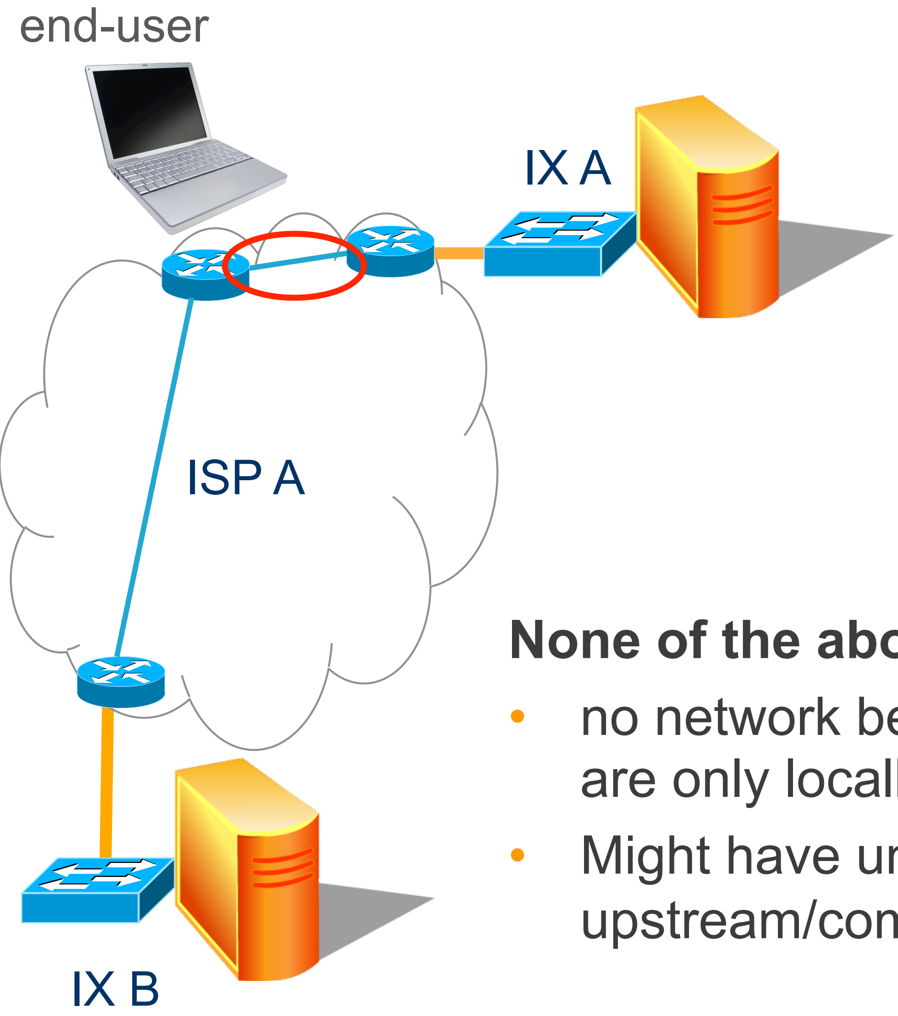
Typical large IX deployment



- Akamai does not have a backbone, each IX instance is independent
- Cluster uses transit to fetch content origin
- Content is served to peers over the IX
- BGP session serves 2 purposes:
  - Route traffic strictly within the local instance
  - Tell our system which prefixes this cluster is allowed to serve
  - New prefixes being picked up by the system can take up to 24hrs

Scenario 1: prefix withdrawal

Attempting to shift traffic to another path



ISP peers with CDN over 2 IXs

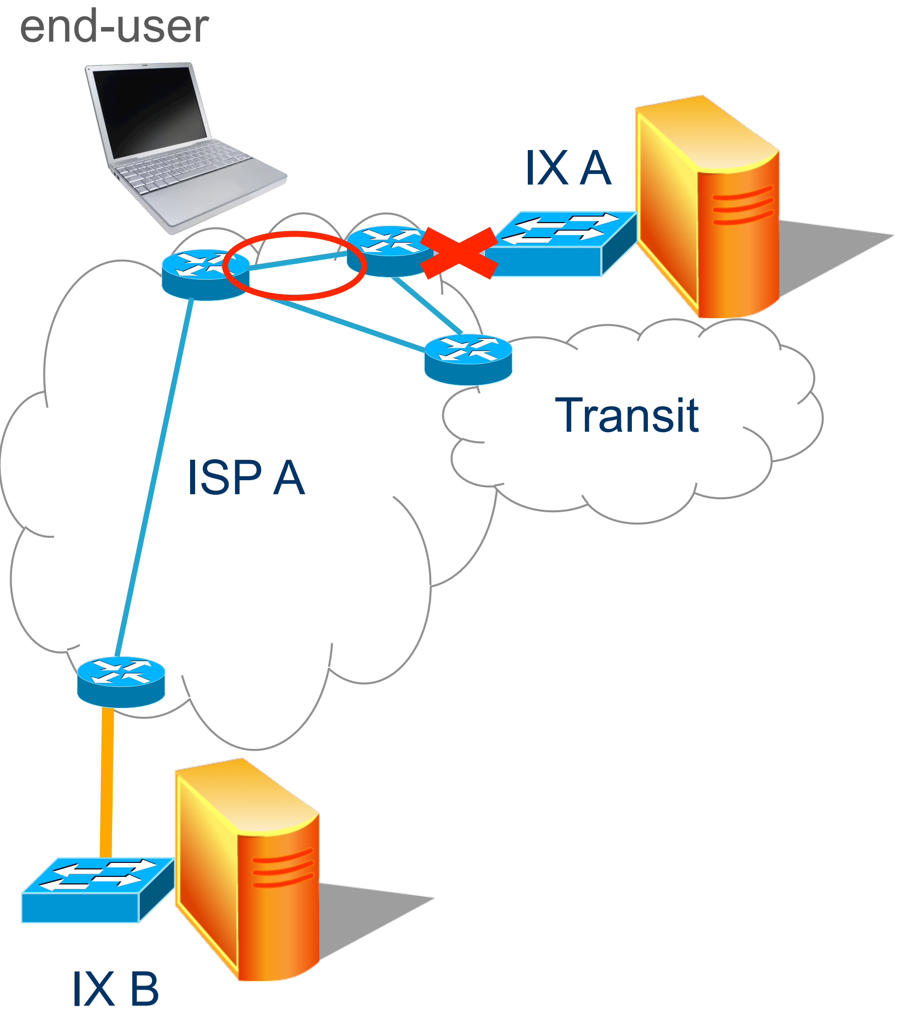
- wants to shift traffic from A to B due to reduce backbone traffic
- Typical BGP based techniques:

1) AS-Path prepending
2) MEDs
3) more/less specific announcements

None of the above have the desired effect!

- no network between clusters, any BGP parameters are only locally relevant
- Might have undesired effect of shifting traffic to a upstream/competitor on the same IX

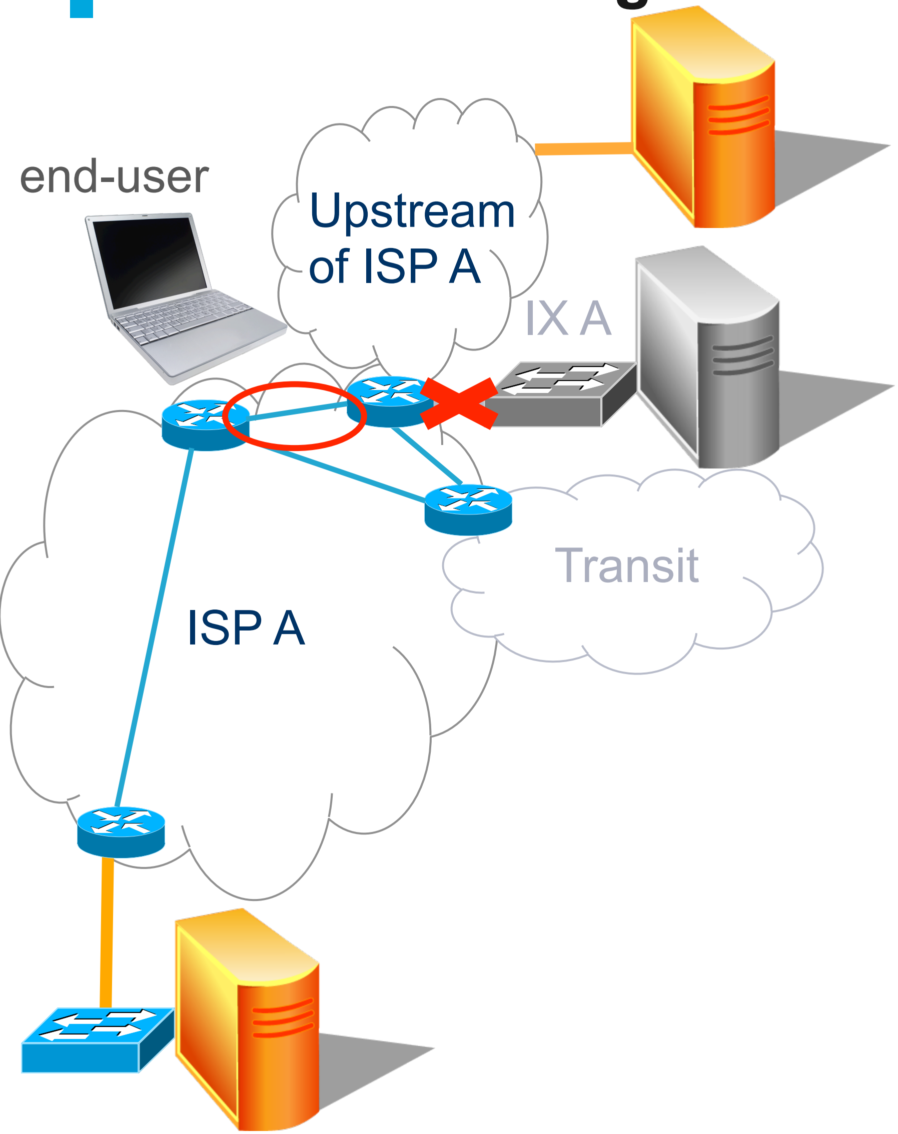
Problem solved…



ISP withdraws some of their prefixes over IX A

- Traffic falls back to transit on the same cluster (provided that still has the best performance)
- this might result in them receiving the traffic in another location in their network so they’re happy

…but not for long



within 24hrs

- The CDN backend system processes the fact that they don't receive their prefixes at the IX A cluster
- Traffic switches to another cluster where they do receive them (this might be one of their upstreams they have a PNI with in the same city)
- Traffic comes back into their network again at the same router!

Issues

- BGP parameters only locally relevant
- Delayed effect of BGP announcements on Mapping System
- CDNs typically see your prefixes in many different locations

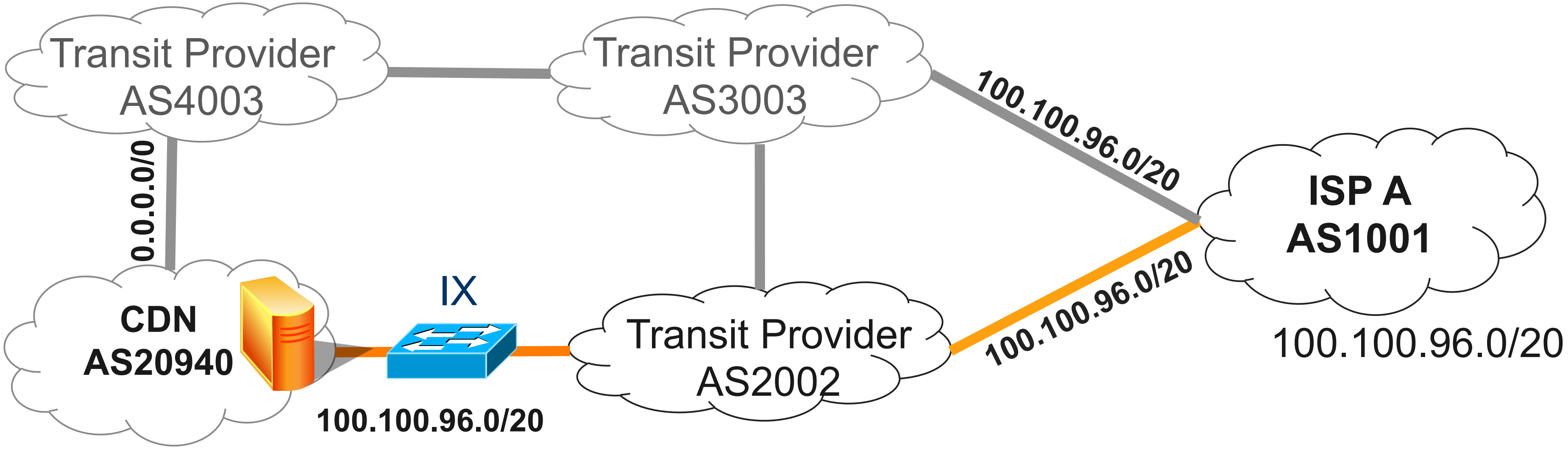
Better solution

- Talk to the CDN if you have issues with their traffic in a specific location
- You can work together to achieve the result you’re looking for
- Get a local embedded cluster

Scenario 2: more specific Route Announcement

Consistent Announcements

- ISP A is multi-homed to Transit Providers AS2002 and AS3003
- Transit Provider AS2002 peers with CDN
- Transit Provider AS3003 does not peer with CDN
- CDN sends traffic to ISP A via Transit Provider AS2002

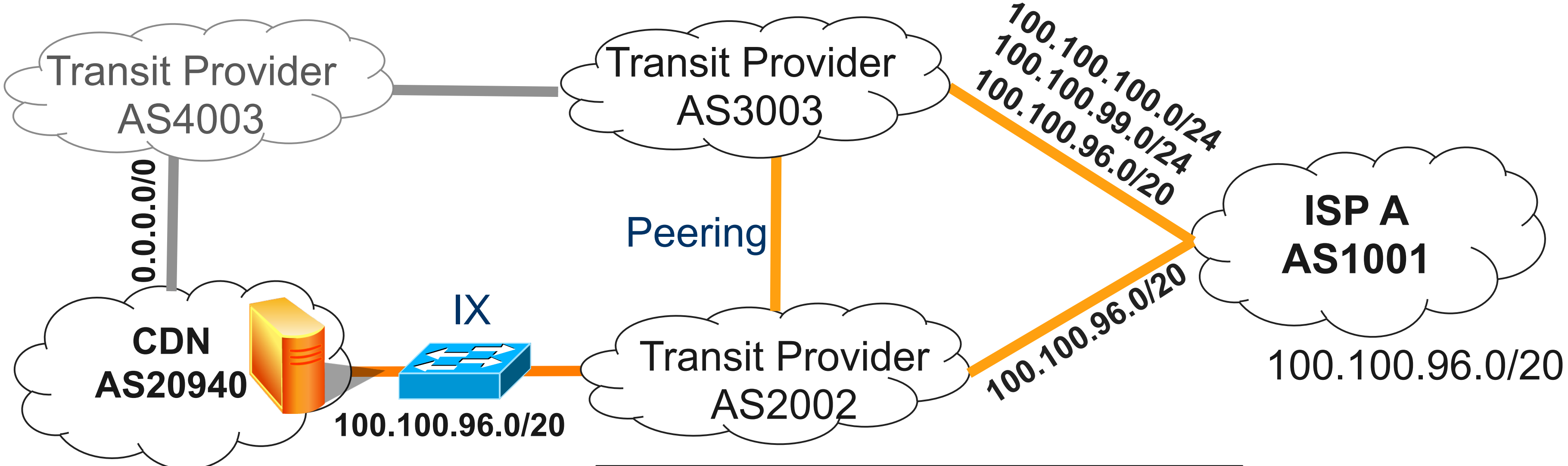


Loadbalancing

- ISP A would like to balance traffic between the two upstream providers
- ISP A prepends, then applies MED to Transit Provider AS2002. This has no effect on CDN traffic.
- Eventually, ISP A de-aggregates the /20 and advertises more specific routes to Transit Provider AS3003
- What will happen?

Loadbalancing works...

- ISP A announces more specific routes to Transit Provider AS3003
- Transit Provider AS3003 announces new /24 to AS2002
- CDN IX router does not have a full-table, so traffic continue route to the /20 of AS2002
- ISP A is happy with the balanced traffic on dual Transit Providers



CDN AS20940 Routing Table

100.100.96.0/20	AS2002 AS1001
0.0.0.0/0	AS4003

AS2002 Routing Table

100.100.100.0/24	AS3003 AS1001
100.100.99.0/24	AS3003 AS1001
100.100.96.0/20	AS1001

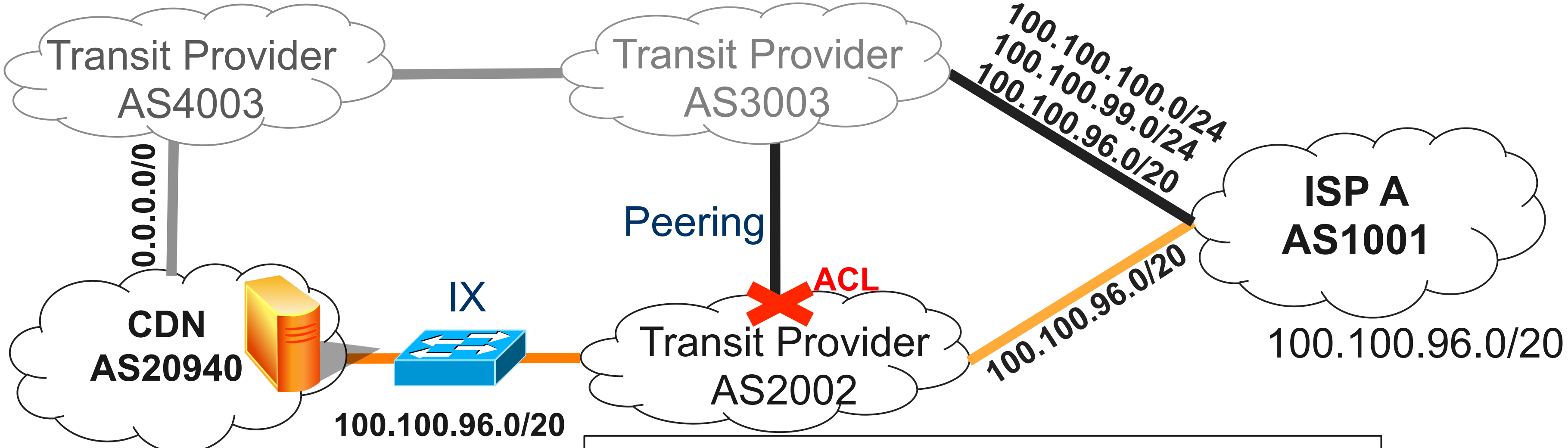
…but

- Lost of revenue for Transit Provider AS2002 even though their peering/backbone is utilized

- What happens if AS2002 does not like the traffic from one peer to the other?

Transit provider filters traffic

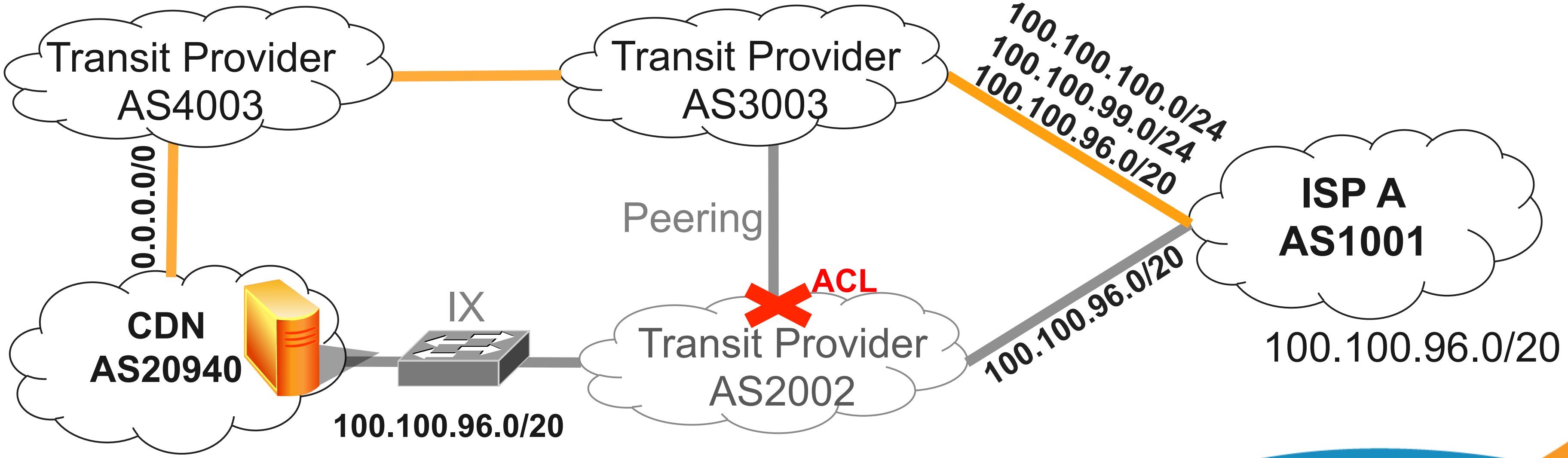
- In order to get rid of traffic between peers, Transit Provider AS2002 implements an ACL on IX port facing AS3003
- Traffic gets blackholed, ISP A's eyeballs don't receive traffic anymore!



```
hostname AS2002-R1
!
interface TenGigabitEthernet1/1
ip access-group 101 out
!
access-list 101 deny ip any 100.100.100.0 0.0.0.255
access-list 101 deny ip any 100.100.99.0 0.0.0.255
access-list 101 permit ip any any
```

Unintended Result

- CDN observes ISP A end-users are unable to access some websites
- CDN stops serving unreliable prefixes received from Transit Provider AS2002, traffic shifts from IX to Transit Provider AS4003
- ISP A can access all websites happily
- Transit Provider AS2002 loses revenue



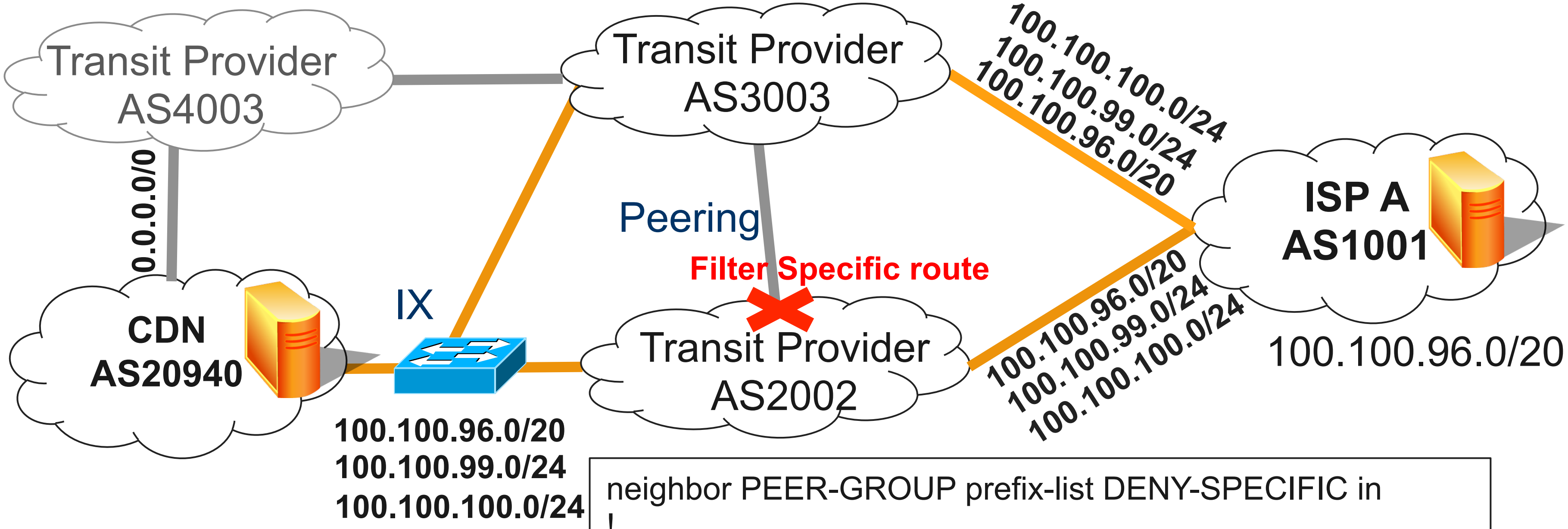
Issues

- Don’t assume a full-table on any device on the internet

- Filtering traffic results in:
  - short term traffic blackholing!
  - long term widthdrawal of traffic resulting in revenue loss

Better solutions

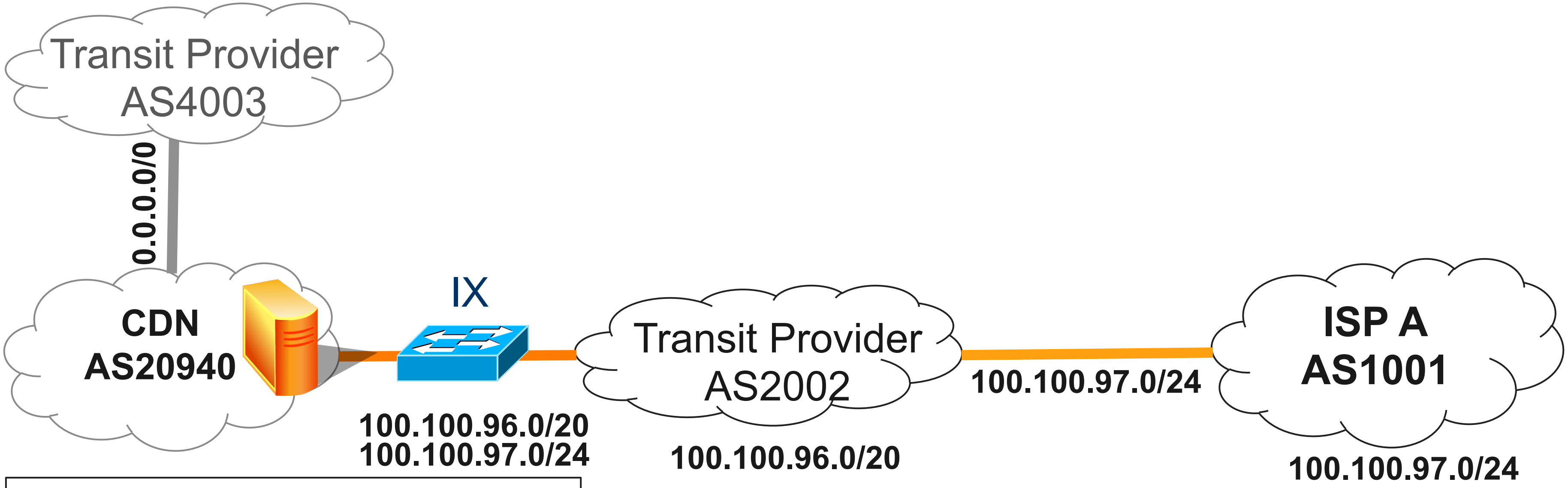
- AS2002 filters the specific prefixes instead of the actual traffic
- work with upstreams and/or CDN for finetuning
- Get Transit Provider AS3003 to peer with CDN directly ;)
- Get a local embedded cluster



```
neighbor PEER-GROUP prefix-list DENY-SPECIFIC in
!
ip prefix-list DENY-SPECIFIC seq 5 deny 100.100.100.0/24
ip prefix-list DENY-SPECIFIC seq 10 deny 100.100.99.0/24
ip prefix-list DENY-SPECIFIC seq 100 permit 0.0.0.0/0 le 32
```

Another variation of this Scenario

- ISP A is single homed to Transit Provider AS2002
- ISP A obtains a /24 from Transit Provider AS2002's address space
- all works well

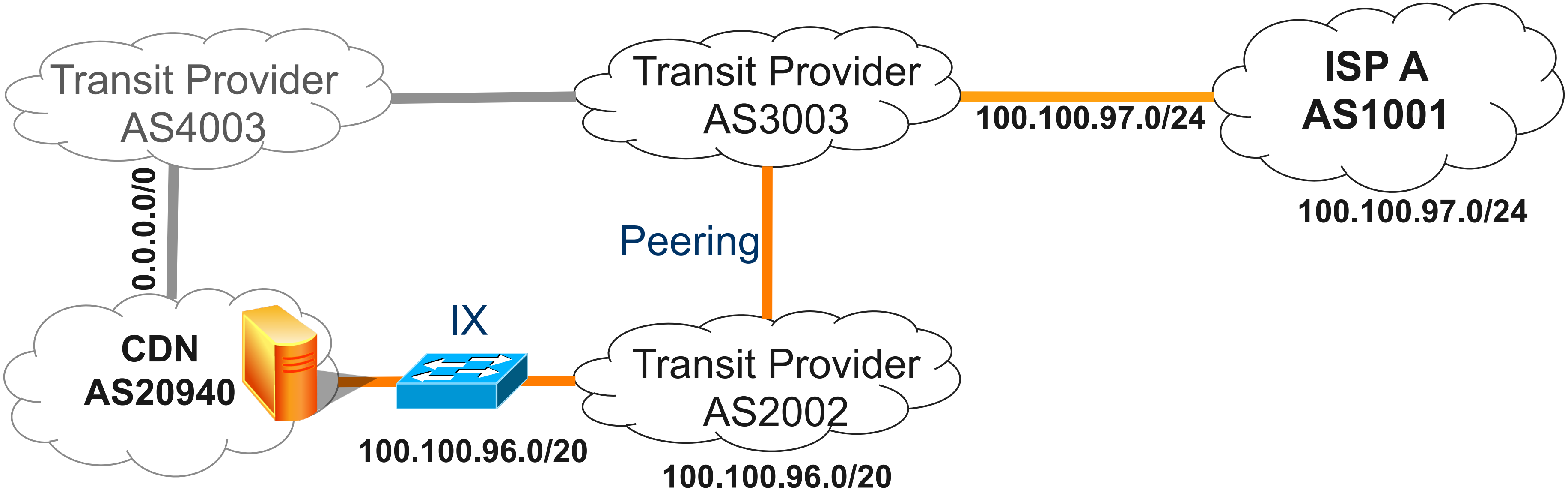


CDN AS20940 Routing Table

100.100.96.0/20	AS2002
100.100.97.0/24	AS2002 AS1001
0.0.0.0/0	AS4003

Provider Change

- ISP A moves to new Transit Provider AS3003, but keeps using his previously assigned prefix 100.100.96.0/24
- CDN keeps serving traffic to ISP A via Transit Provider AS2002 due to the /20 being received there



CDN AS20940 Routing Table

100.100.96.0/20	AS2002
0.0.0.0/0	AS4003

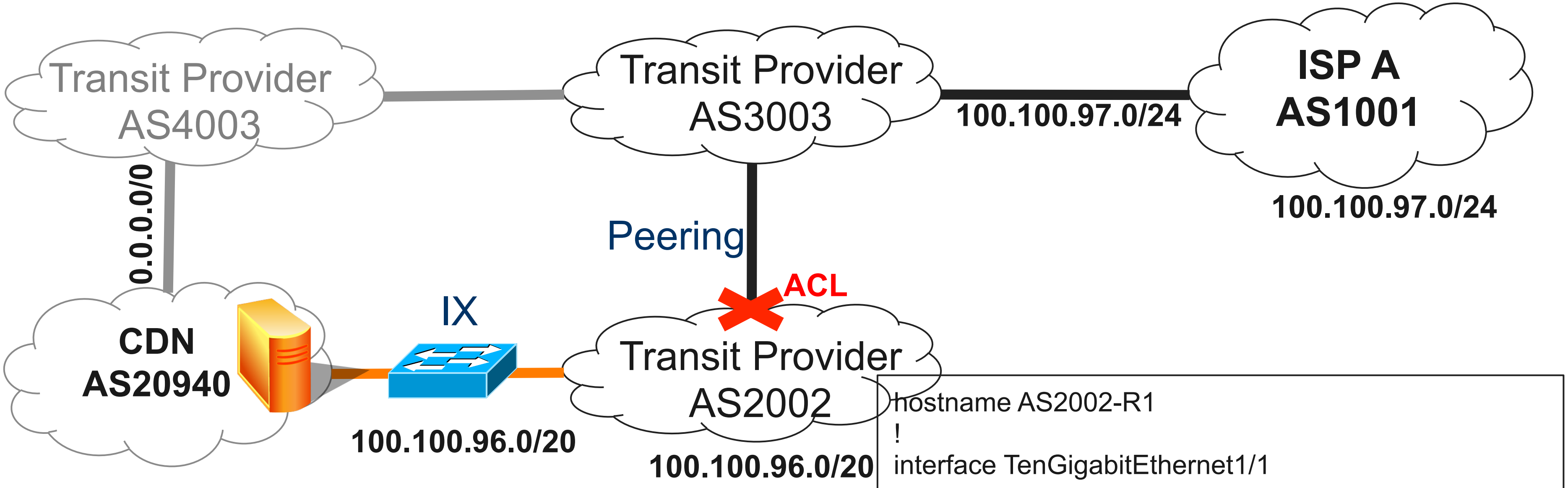
…but

- Lost of revenue for Transit Provider AS2002 even though their peering/backbone is utilized

- What happens if AS2002 does not like the traffic from one peer to the other?

Transit provider filters traffic

- In order to get rid of traffic between peers, Transit Provider AS2002 implements an ACL on IX port facing AS3003
- Traffic gets blackholed, ISP A’s eyeballs don’t receive traffic anymore!



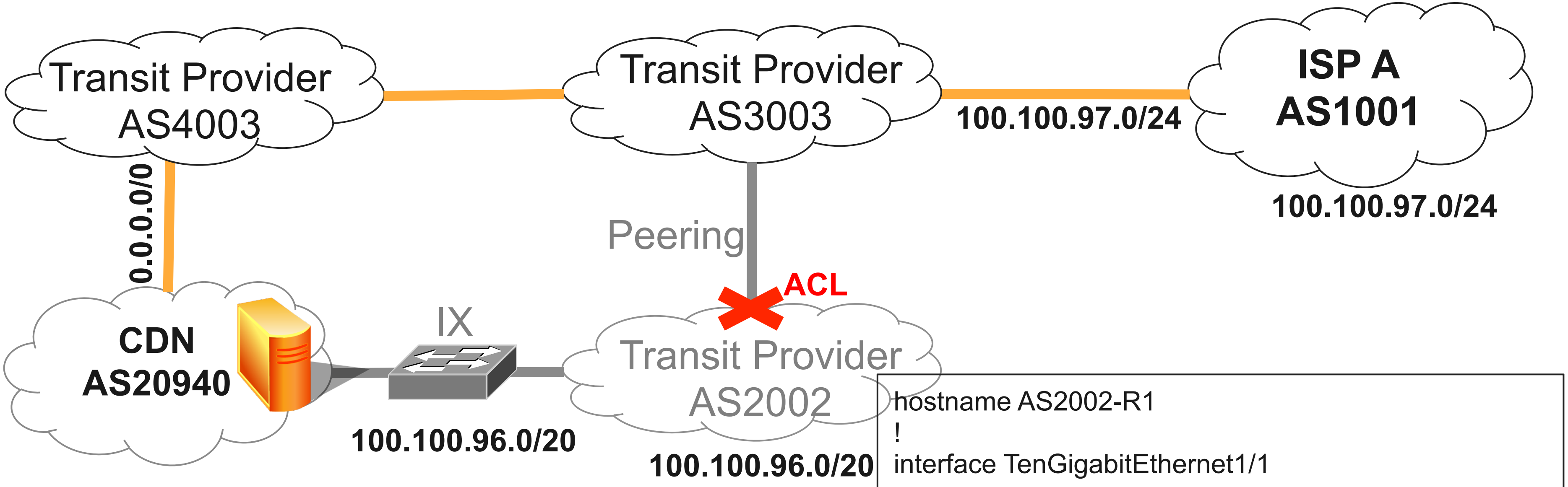
```
hostname AS2002-R1
!
interface TenGigabitEthernet1/1
ip access-group 101 out
!
access-list 101 deny ip any 100.100.97.0 0.0.0.255
access-list 101 permit ip any any
```

CDN AS20940 Routing Table

100.100.96.0/20	AS2002
0.0.0.0/0	AS4003

Unintended Result

- CDN observes ISP A end-users are unable to access some websites
- CDN stops serving unreliable prefixes received from Transit Provider AS2002, traffic shifts from IX to Transit Provider AS4003
- ISP A can access all websites happily
- Transit Provider AS2002 loses revenue



CDN AS20940 Routing Table

~~100.100.96.0/20~~	~~AS2002~~
0.0.0.0/0	AS4003

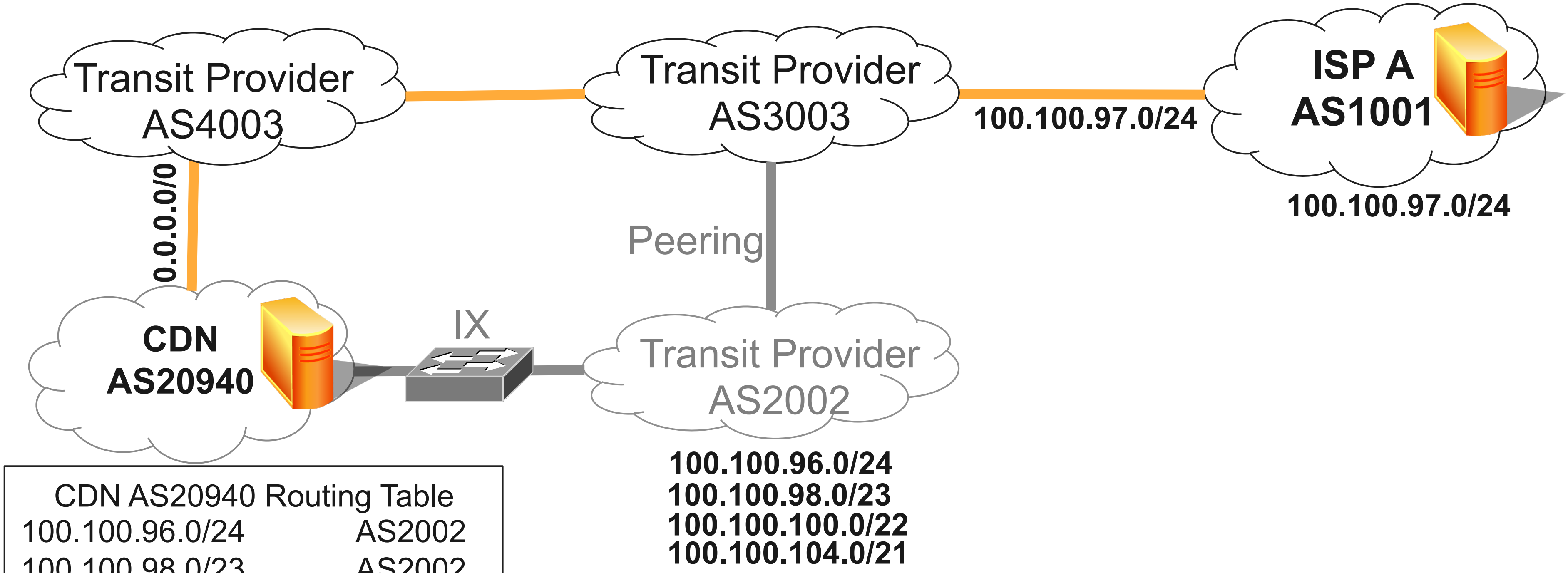
```
hostname AS2002-R1
!
interface TenGigabitEthernet1/1
ip access-group 101 out
!
access-list 101 deny ip any 100.100.97.0 0.0.0.255
access-list 101 permit ip any any
```

Issues

- Don’t assume a full-table on any device on the internet
- If you do announce a prefix others expect you to be able to serve traffic to all of it
- Don’t allow customers to use your PA space as PI
- Filtering traffic results in:
  - short term traffic blackholing!
  - long term widthdrawal of traffic resulting in revenue loss

Better solutions

- Deaggregate the /20 if you can't/won't serve all of it
- Only announce address space you will serve traffic to
- Get a local embedded cluster



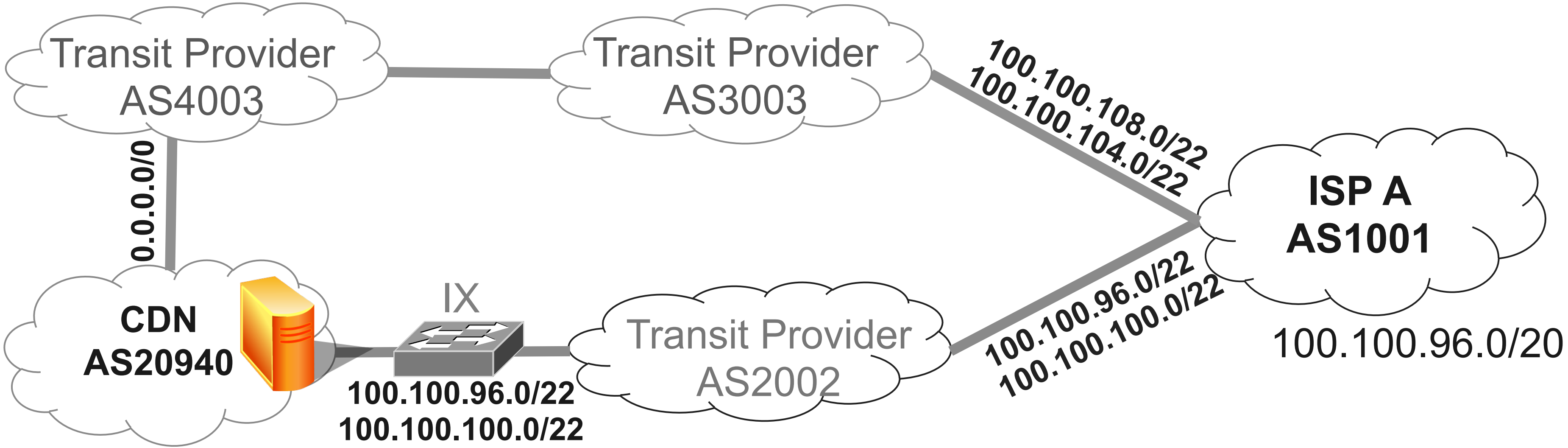
CDN AS20940 Routing Table

100.100.96.0/24	AS2002
100.100.98.0/23	AS2002
100.100.100.0/22	AS2002
100.100.104.0/21	AS2002
0.0.0.0/0	AS4003

Scenario 3: Split Route Announcement

Split announcement

- ISP A is multi-homed to Transit Providers AS2002 and AS3003
- Transit Provider AS2002 peers with CDN
- Transit Provider AS3003 does not peer with CDN
- ISP A announces different prefix to different ISP
- ISP A can access full internet

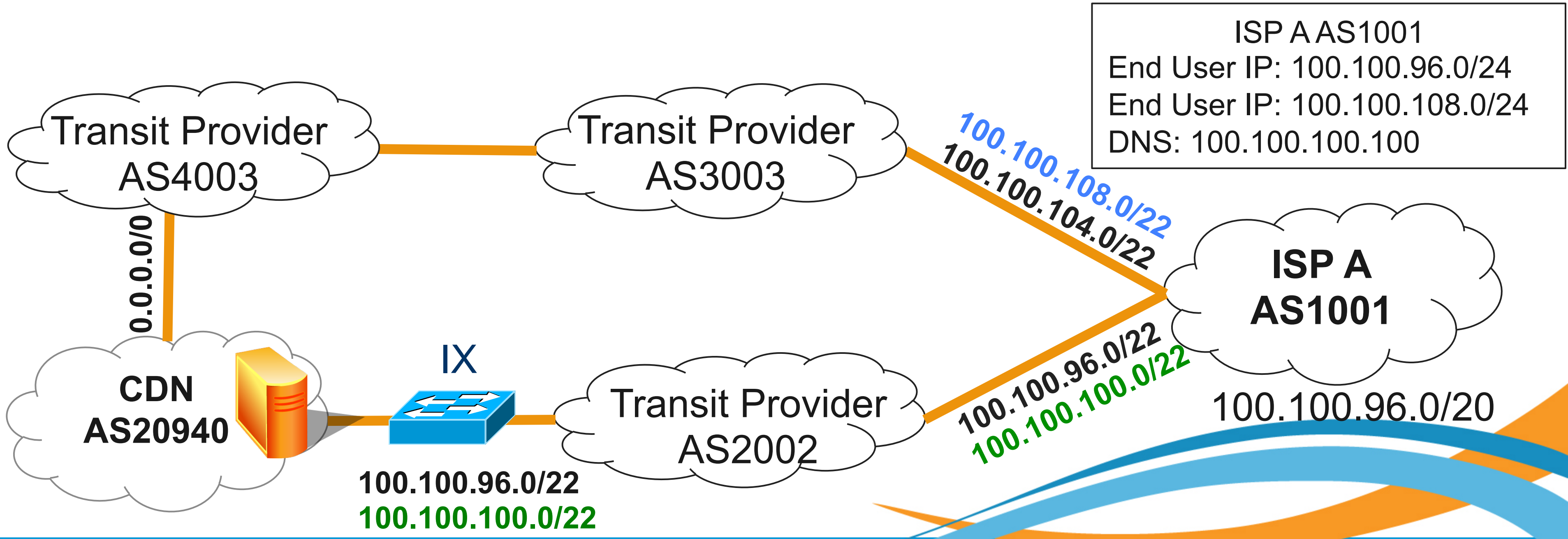


CDN AS20940 Routing Table

100.100.96.0/22	AS2002 AS1001
100.100.100.0/22	AS2002 AS1001
0.0.0.0/0	AS4003

Split announcement

- End Users are using IP Addresses 100.100.96.0/22, 100.100.100.0/22, 100.100.104.0/22, 100.100.108.0/22
- End Users are using ISP A NS 100.100.100.100
- CDN receives the NS Prefix 100.100.100.0/22 from AS2002 and maps the traffic for ISP A to this cluster
- 100.100.96.0/22 100.100.100.0/22 traffic is routed via AS2002 while 100.100.104.0/22 100.100.108.0/22 traffic falls back to default route via AS4003, AS3003

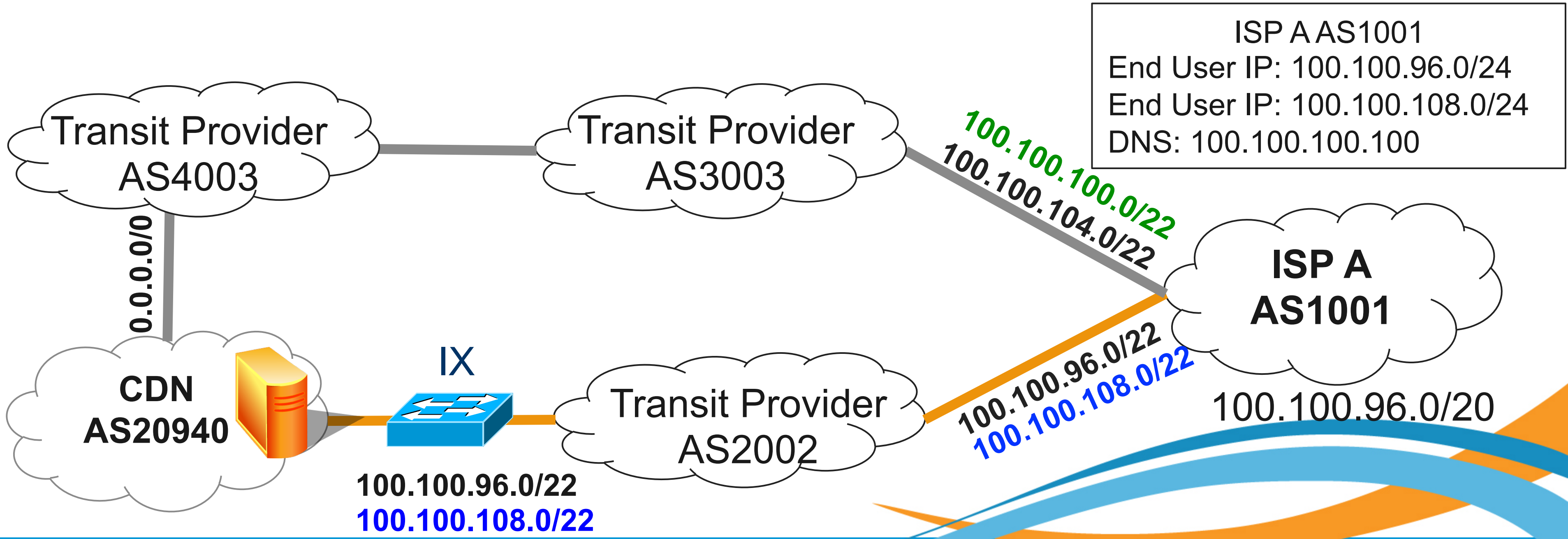


Differing performance

- This can work perfectly fine
- But the path via the transit providers AS4003 & AS3003 might not be as good as the direct peering, 100.100.100.108.0/22 end users could have significantly worse performance
- What will ISP A do if the user complain?

Problem solved…

- ISP A swaps the route announcements
- Both 100.100.96.0/22 and 100.100.108.0/22 are routed via AS2002 and end-users have the same performance
- The end-user is happy and closes the ticket



ISP A AS1001
End User IP: 100.100.96.0/24
End User IP: 100.100.108.0/24
DNS: 100.100.100.100

…but

24hrs later:

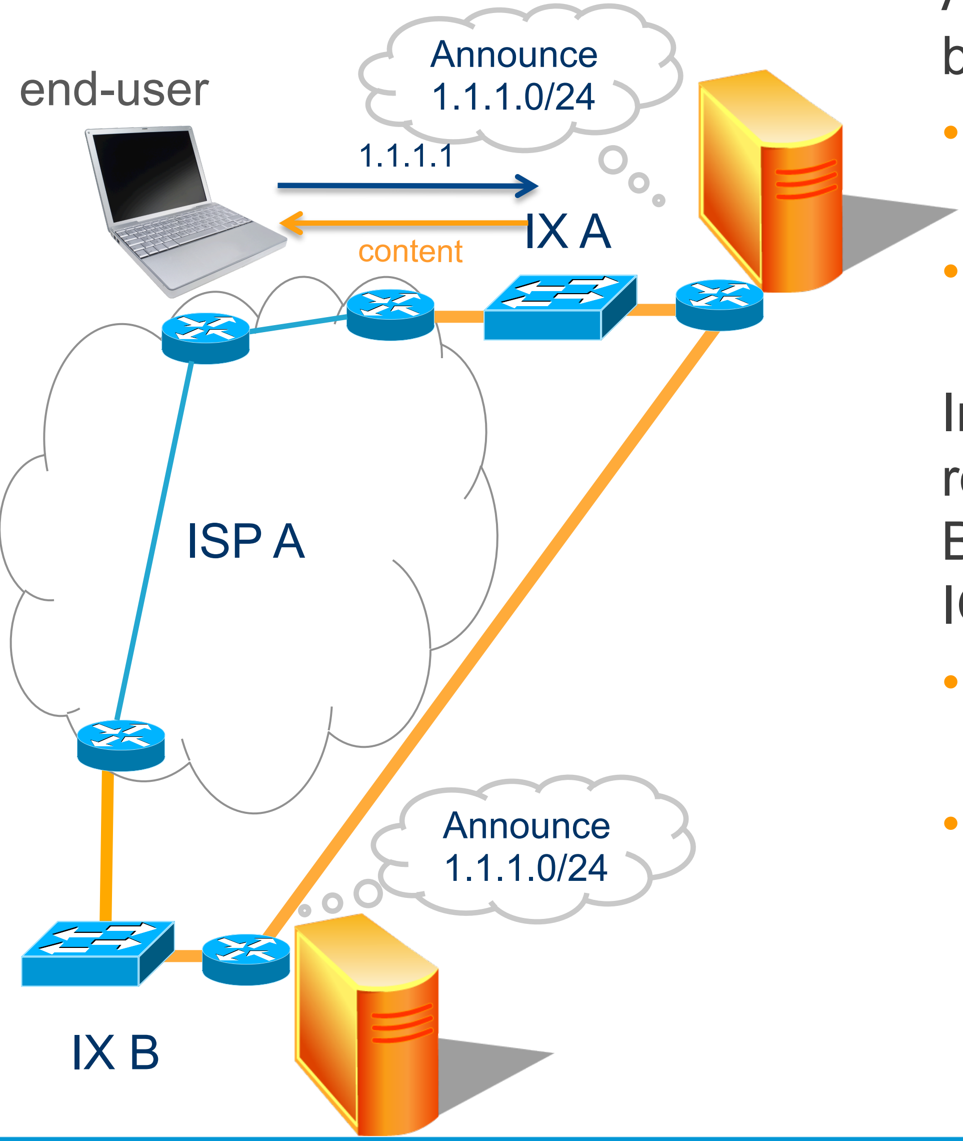
- CDN no longer receives NS prefix 100.100.100.0/22 from AS2002
- CDN maps the traffic of ISP A to Cluster B (where they see AS3003's prefixes) instead of Cluster A (which only peers with AS2002)
- ISP A will receive the traffic from a completely different source potentially all via AS3003 now negating all the TE efforts

DO NOT split nameserver and end-user prefixes when traffic engineering

Akamai
FASTER FORWARD

Scenario 4: Anycast examples

Typical anycast CDN setup



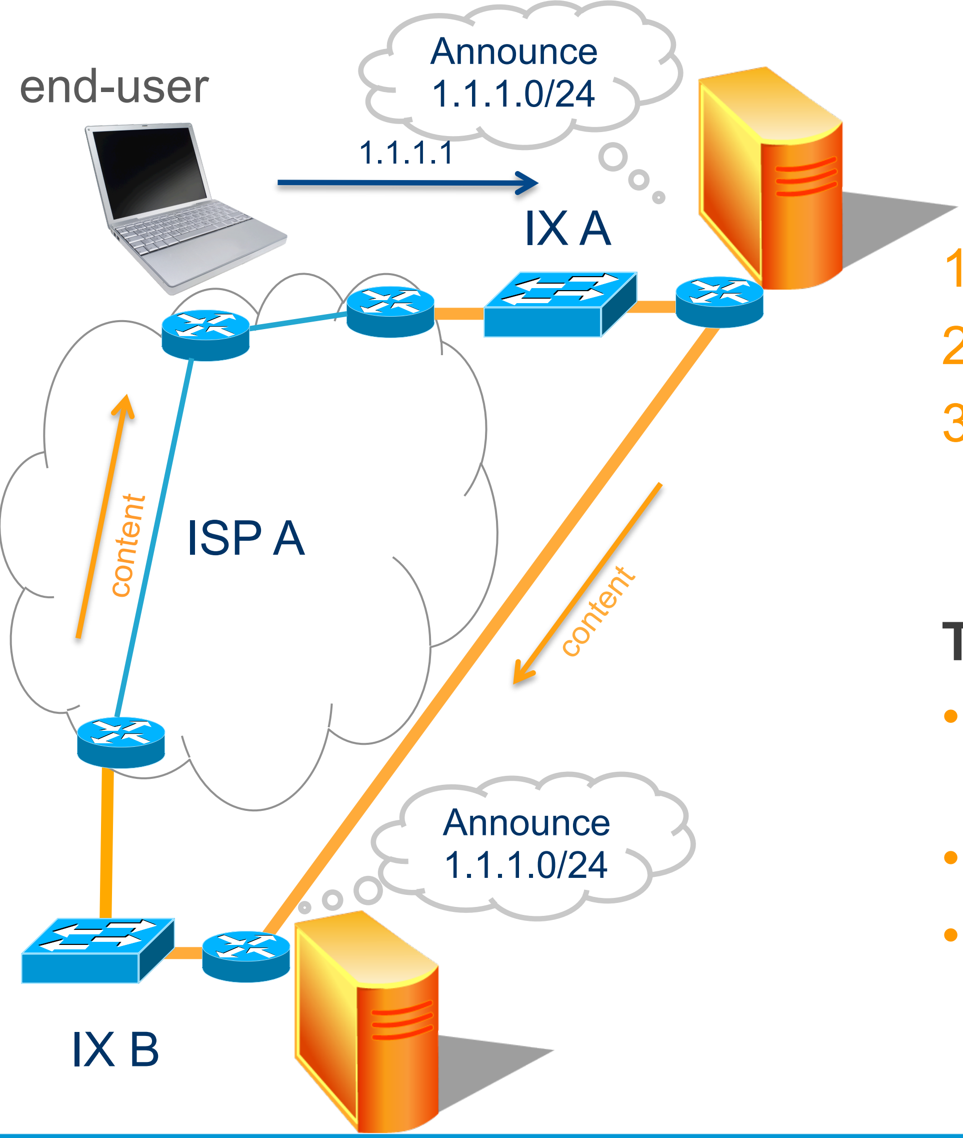
Anycast based CDNs usually have a backbone

- Announce same set of prefixes at each location
- Serve traffic from the cluster that receives the inbound request

In the best case (symmetrical routing) traffic both ways follows BGP logic (shortest as path, lowest IGP metric)

- Outbound from ISP goes to nearest peering with CDN
- Inbound to ISP comes from nearest peering

TE effects on anycast CDNs



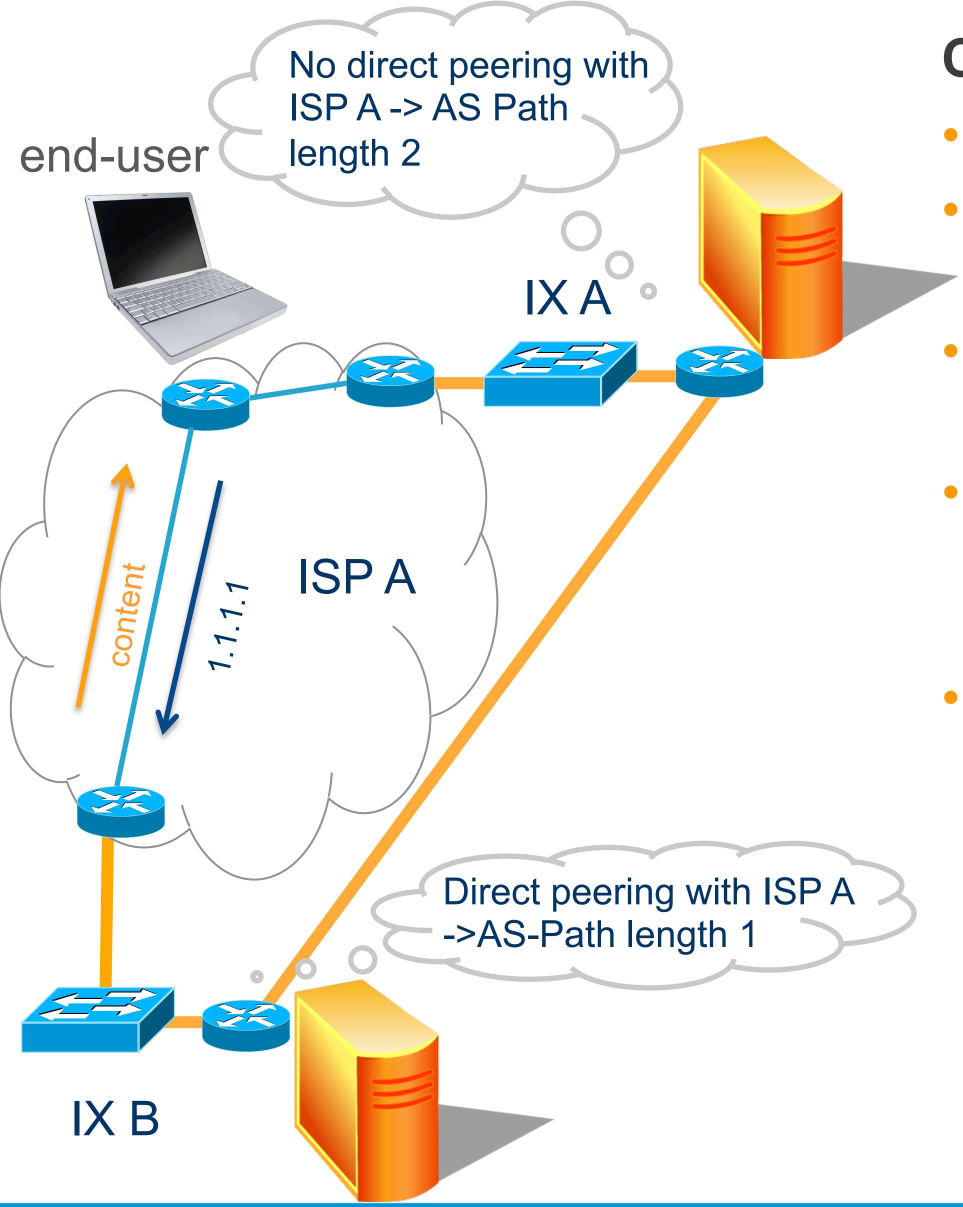
- Typical BGP based techniques work on anycast CDNs for inbound traffic to the ISP

1) AS-Path prepending
2) MEDs
3) more/less specific announcements

Take outbound path into account!

- Traffic still gets served from the cluster that receives the request
- Routes the ‘long way round’
- Worse performance than if it was served from IX B

TE effects on anycast CDNs


No direct peering with ISP A -> AS Path length 2
end-user
IX A
ISP A
content
1.1.1.1
Direct peering with ISP A ->AS-Path length 1
IX B

Chose Peering locations carefully

- Routing follows BGP parameters
- This is fine if IX A & B are close to you
- What if IX B is on another continent?
- Gets more complex when peering relations are indirect (Transit Providers in between)
- Be careful with route-servers (as you might unintentionally create this scenario)

Scenario 4: (Attempted) Content Filtering

Content filtering

ISPs do receive requests from Government organizations to filter specific content

The default action is often to block a specific source IP

If this content is hosted by a CDN this will not do what you expect!

what it WILL DO:

blackhole **random** content to your end-users

get your cluster **suspended** because of observed loss and all traffic served from upstream

what it will NOT:

filter any **specific** content

Summary

- Standard BGP traffic engineering will not have the expected results
- Changes in announcements may have a delayed effect
- Where mapping is based on NS, splitting nameserver and end-user prefixes over different providers will have unexpected effects
- Not all clusters have a full table
  - splitting more specific announcements over different links can cause unintended behavior
  - Announcing prefixes with holes results in blackholing traffic

- Talk to your CDN partners for finetuning traffic
- DO NOT filter traffic by IP

Questions?

Matt Jansen mj@akamai.com

as20940.peeringdb.com