

# BGP and Traffic Engineering with Akamai

Owen DeLong  
Akamai Technologies  
LACNIC 2016, San Jose, CR



# AGENDA



- Akamai Intelligent Platform
- Peering with Akamai
- Traffic Engineering
- Summary
- Q&A

# The Akamai Intelligent Platform



Highly distributed, deeply deployed on-demand computing platform that serves any kind of web traffic and applications

## The Akamai Intelligent Platform:

**200,000+**  
Servers

**2,000+**  
Locations

**1,300+**  
Networks

**1100+**  
Cities

**130+**  
Countries



### Typical daily traffic:

- More than **2 trillion** requests served
- Delivering over **33 Terabits/second**
- **15-30%** of all daily web traffic

# Basic Technology

Akamai mapping



# How CDNs Work



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

There are 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 resolving name server of the provider it receives the request from, and replies with the optimal server



# 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

# How Akamai CDN Work



## Example of Akamai mapping

- Notice the different A records for different locations:

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

# Peering with Akamai





# Why Akamai Peer with ISPs



## Performance & Redundancy

- Removing intermediate AS hops gives higher peak traffic for same demand profile

## Burstability

- During large events, having direct connectivity to multiple networks allows for higher burstability than a single connection to a transit provider

## Peering reduces costs

## Backup for on-net servers

- If there are servers on-net, the peering can act as a backup during downtime and overflow
- Allows serving different content types

# Why ISPs peer with Akamai



## Performance

- ISP's end-users benefit from direct connectivity

## Competitive Advantages

- improving performance over competitors
- additional revenue from downstreams

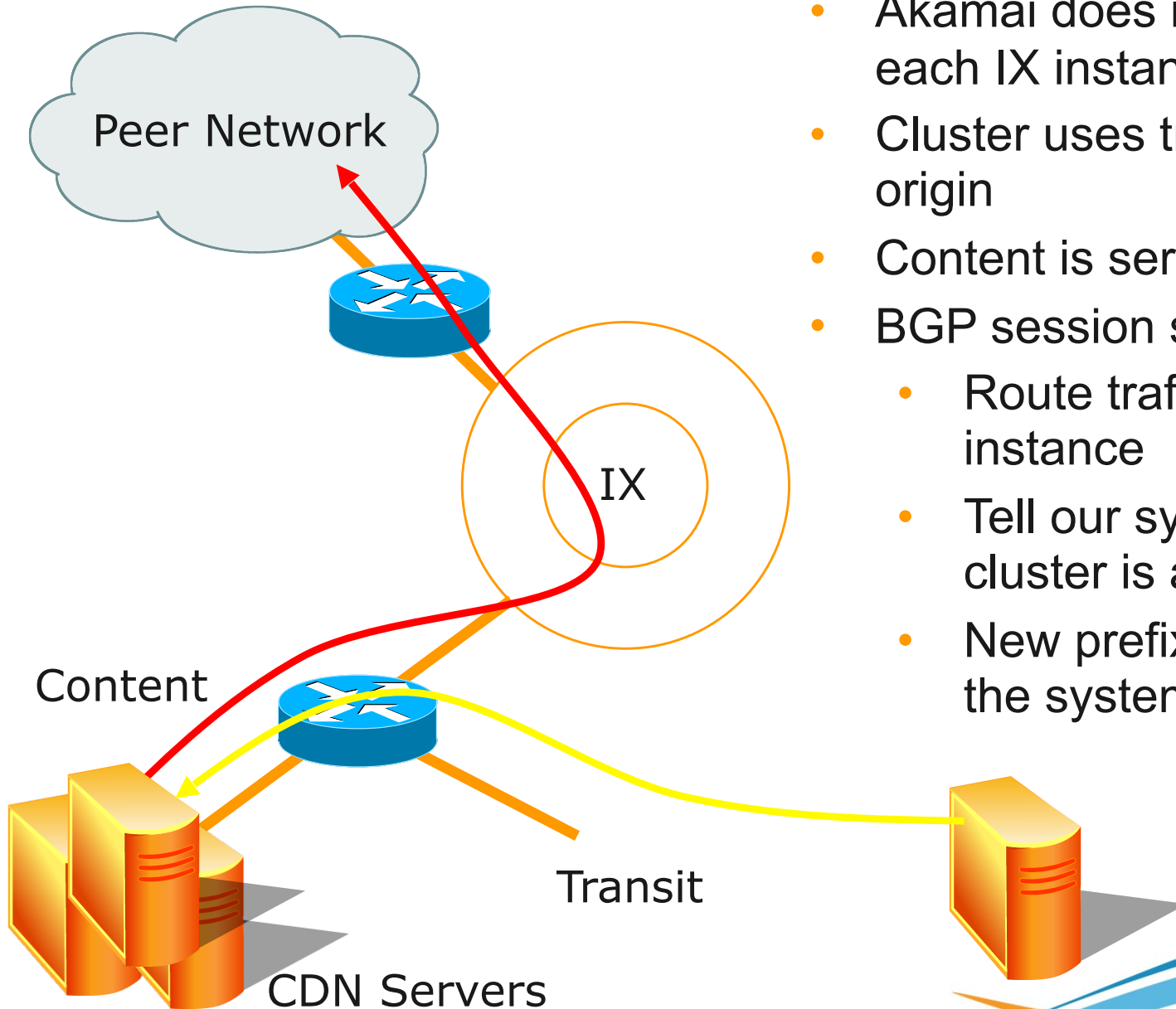
## Cost Reduction

- Save on transit bill and potential backbone costs

## Redundancy

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

# How Akamai use IXes



- 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

## How Akamai use IXes



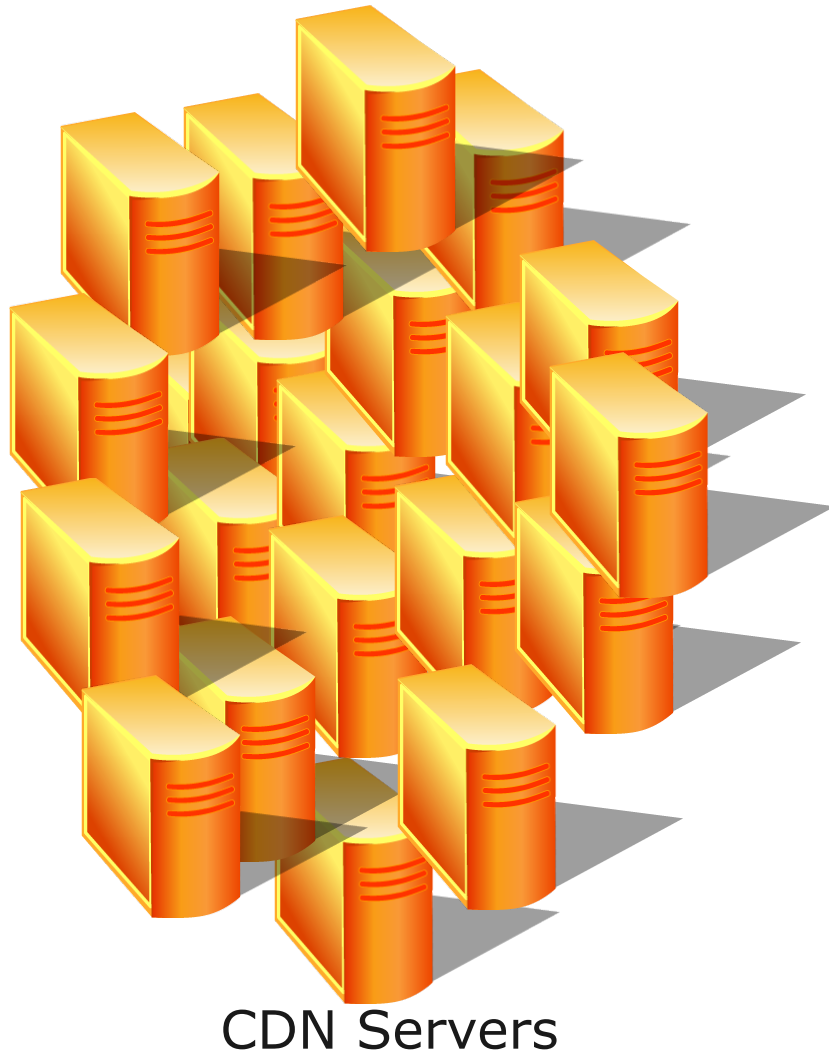
Akamai usually does not announce large blocks of address space because no one location has a large number of servers

- It is not uncommon to see a few small prefixes (/22, /23) from Akamai at an IX

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

- How many web servers does it take to fill a 10G these days?

# Why don't I get all the Akamai content via the Peering?



- No single cluster can accommodate all Akamai content
- Clusters get more efficient with with size
- Some content requires specialized servers only present in Infrastructure clusters
- Some content is only present in specific geographies
- Do you want to host Akamai Cluster?



# After Peering With Akamai....

DO and DON'T's of Traffic Engineering



# Typical BGP-based TE Techniques



# AS Path Prepending



- **Before**

```
Akamai Router#sh ip b 100.100.100.100
```

```
BGP routing table entry for 100.100.100.0/20, version Paths: (1 available, best #1, table Default-IP-Routing-Table)
```

```
Multipath: eBGP
```

```
Advertised to update-groups:
```

```
2 7
```

```
4635 1001
```

```
202.40.161.1 from 202.40.161.1 (202.40.161.1)
```

- **After**

```
Akamai Router#sh ip b 100.100.100.100
```

```
BGP routing table entry for 100.100.100.0/20, version  
Paths: (1 available, best #1, table Default-IP-Routing-Table)
```

```
Multipath: eBGP
```

```
Advertised to update-groups:
```

```
2 7
```

```
4635 1001 1001 1001 1001
```

```
202.40.161.1 from 202.40.161.1 (202.40.161.1)
```

# MED



- **Before**

Akamai Router#sh ip b 100.100.100.100

BGP routing table entry for 100.100.100.0/20, version Paths: (1 available, best #1, table Default-IP-Routing-Table)

Multipath: eBGP

Advertised to update-groups:

2 7

4635 1001

202.40.161.1 from 202.40.161.1 (202.40.161.1)

Origin IGP, **metric 0**, localpref 100, valid, external, best

- **After**

Akamai Router#sh ip b 100.100.100.100

BGP routing table entry for 100.100.100.0/20, version Paths: (1 available, best #1, table Default-IP-Routing-Table)

Multipath: eBGP

Advertised to update-groups:

2 7

4635 1001

202.40.161.1 from 202.40.161.1 (202.40.161.1)

Origin IGP, **metric 1000**, localpref 100, valid, external, best

# More Specific Route



- **Before**

```
Akamai Router#sh ip b 100.100.100.100
BGP routing table entry for 100.100.96.0/20, version
Paths: (1 available, best #1, table Default-IP-Routing-Table)
Multipath: eBGP
  Advertised to update-groups:
    2      7
  4635 1001
    202.40.161.1 from 202.40.161.1 (202.40.161.1)
```

- **After**

```
Akamai Router#sh ip b 100.100.100.100
BGP routing table entry for 100.100.100.0/24, version Paths: (1 available, best #1, table
Default-IP-Routing-Table)
Multipath: eBGP
  Advertised to update-groups:
    2      7
  4635 1001
    202.40.161.1 from 202.40.161.1 (202.40.161.1)
```



These will not have the desired effects



# Why doesn't it have the usual effect?



- Akamai uses Mapping, on top of the BGP routing
- Akamai Mapping is different from BGP routing
- Akamai uses 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

# Typical Scenarios in Traffic Engineering





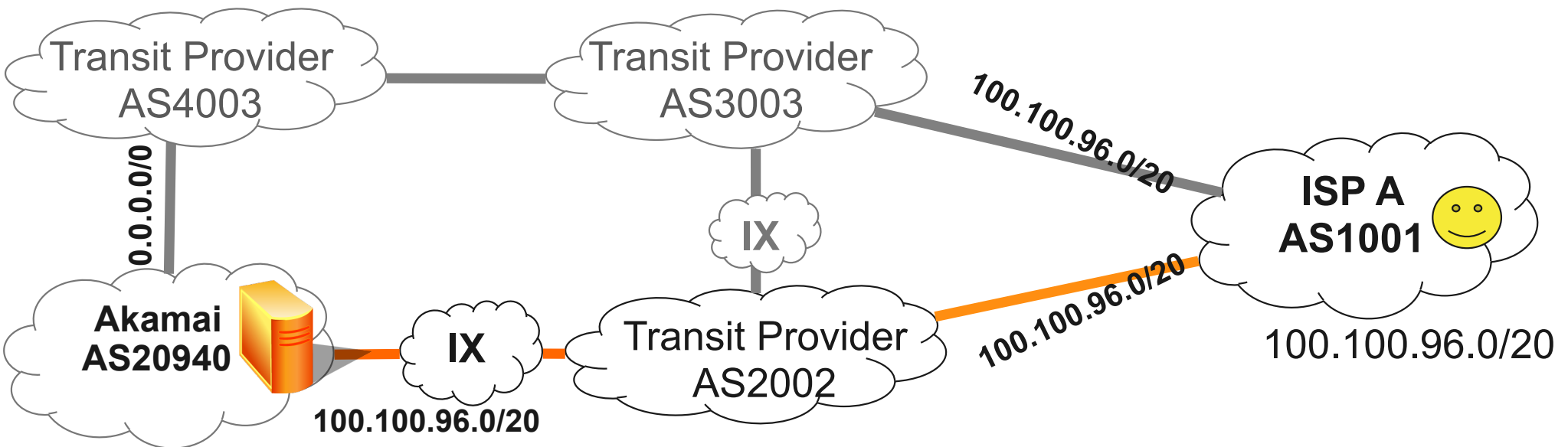
# Scenario 1: Inconsistent Route Announcement



# Consistent Route Announcements



- ISP A is multi-homed to Transit Provider AS2002 and AS3003
- Transit Provider AS2002 peer with Akamai
- Transit Provider AS3003 does not peer with Akamai
- Akamai always sends traffic to ISP A via Transit Provider AS2002





# Load-Balancing

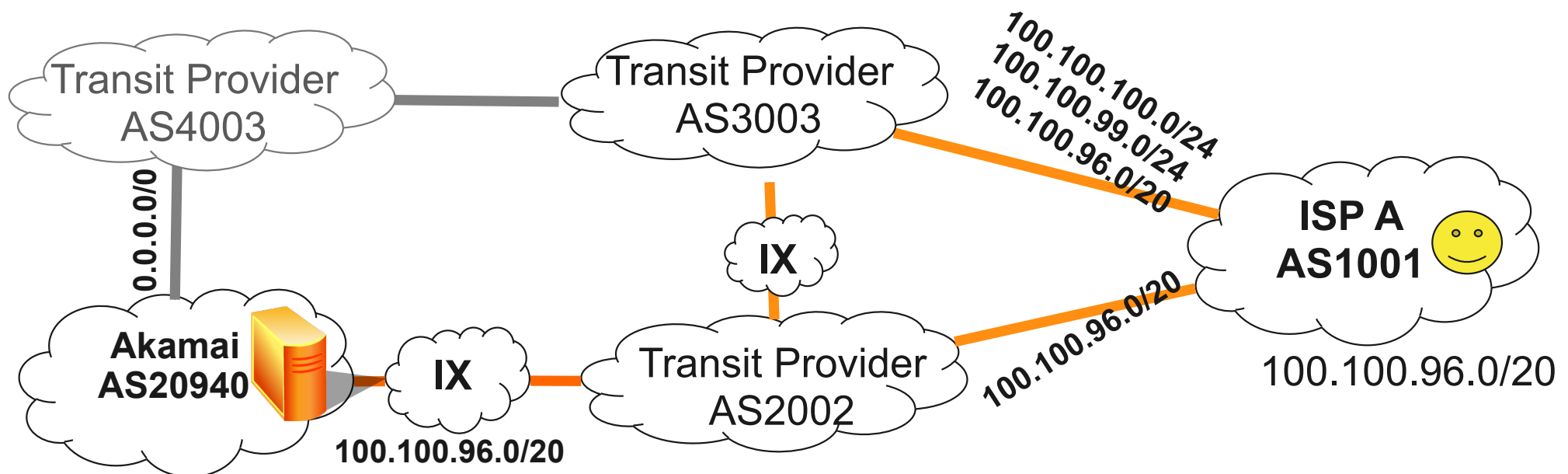


- ISP A would like to balance the traffic between two upstream providers
- ISP A prepends, applies MED to Transit Provider AS2002. Unfortunately, no effect on Akamai traffic.....
- Eventually, ISP A de-aggregates the /20 and advertises more specific routes to Transit Provider AS3003
- What will happen?

# ISP A Load Balance the Traffic Successfully



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



100.100.96.0/20	AS2002 AS1001
0.0.0.0/0	AS4003

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

## What is the problem?

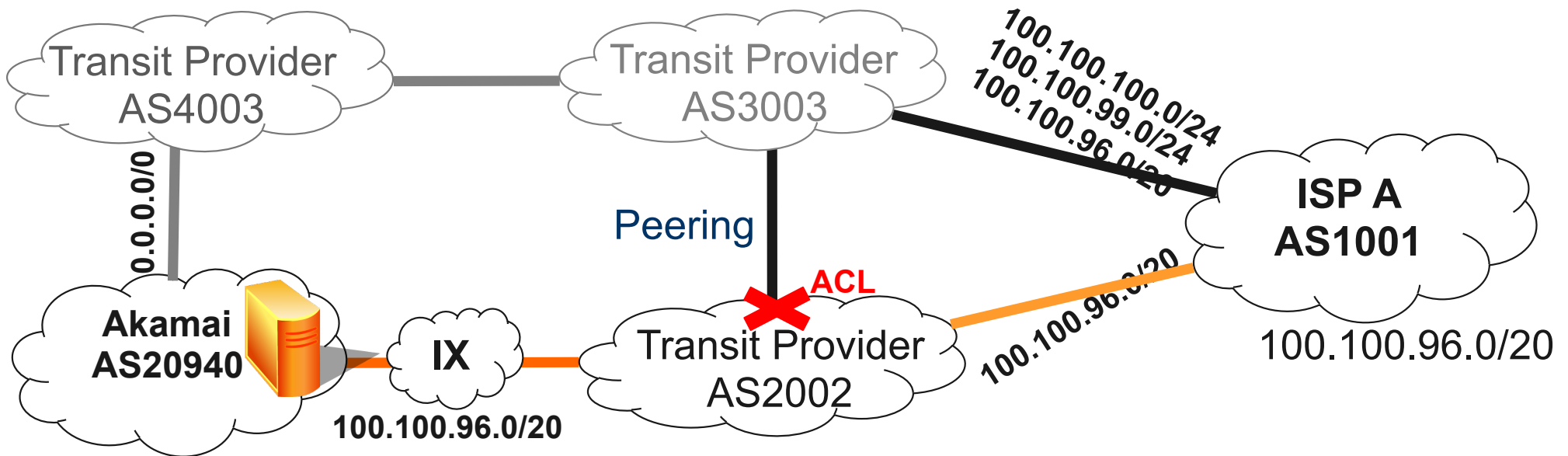


- Lost of revenue for Transit Provider AS2002 even though their peering/backbone is utilized
- What could happen if AS2002 does not like the peer-to-peer traffic?

# Transit provider filters traffic



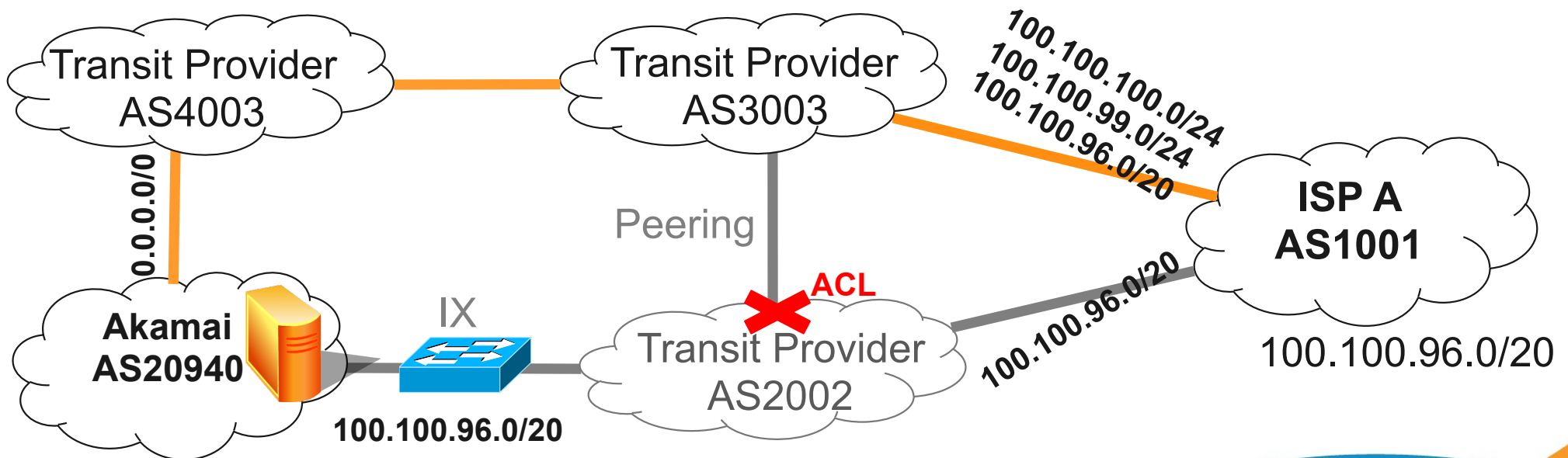
- In order to get rid of traffic between peers, Transit Provider AS2002 implements an ACL on IX port facing AS3003
- Traffic gets blackholed, ISP A's eyeballs don't receive traffic anymore!



# Unintended Result



- Akamai observes ISP A end-users are unable to access some websites
- Akamai 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 withdrawal of traffic resulting in revenue loss



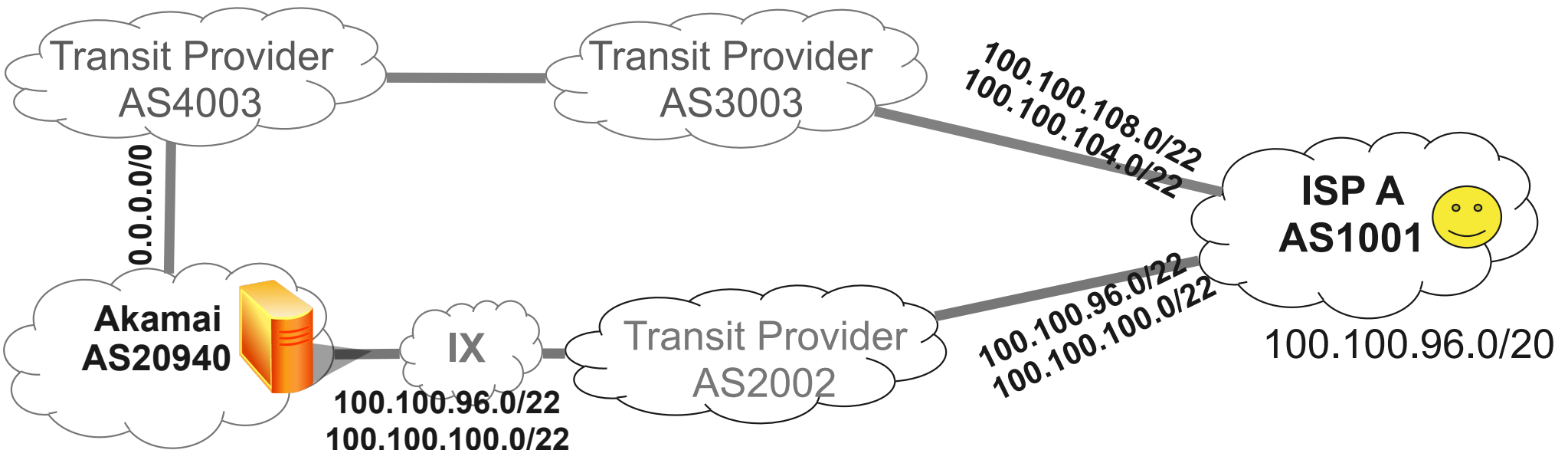
# Scenario 2: Incomplete Route Announcement



# Incomplete Route Announcement



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

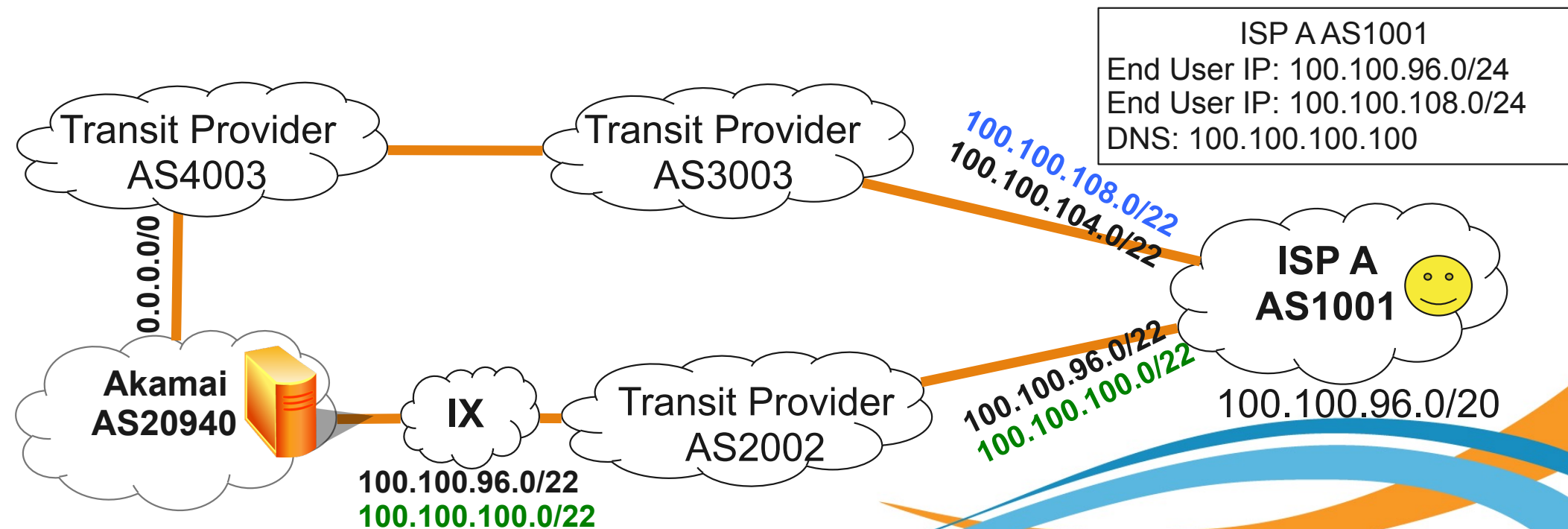


100.100.96.0/22	AS2002 AS1001
100.100.100.0/22	AS2002 AS1001
0.0.0.0/0	AS4003

# How will the traffic be routed to ISP A end users?



- End Users are using IP Address of 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 DNS Server 100.100.100.100
- Akamai receives the DNS Prefix 100.100.100.0/22 from AS2002, so it maps the traffic of ISP A to this cluster
- 100.100.96.0/22 100.100.100.0/22 traffic is routed to AS2002 while 100.100.104.0/22 100.100.108.0/22 traffic is routed to AS3003 by default route



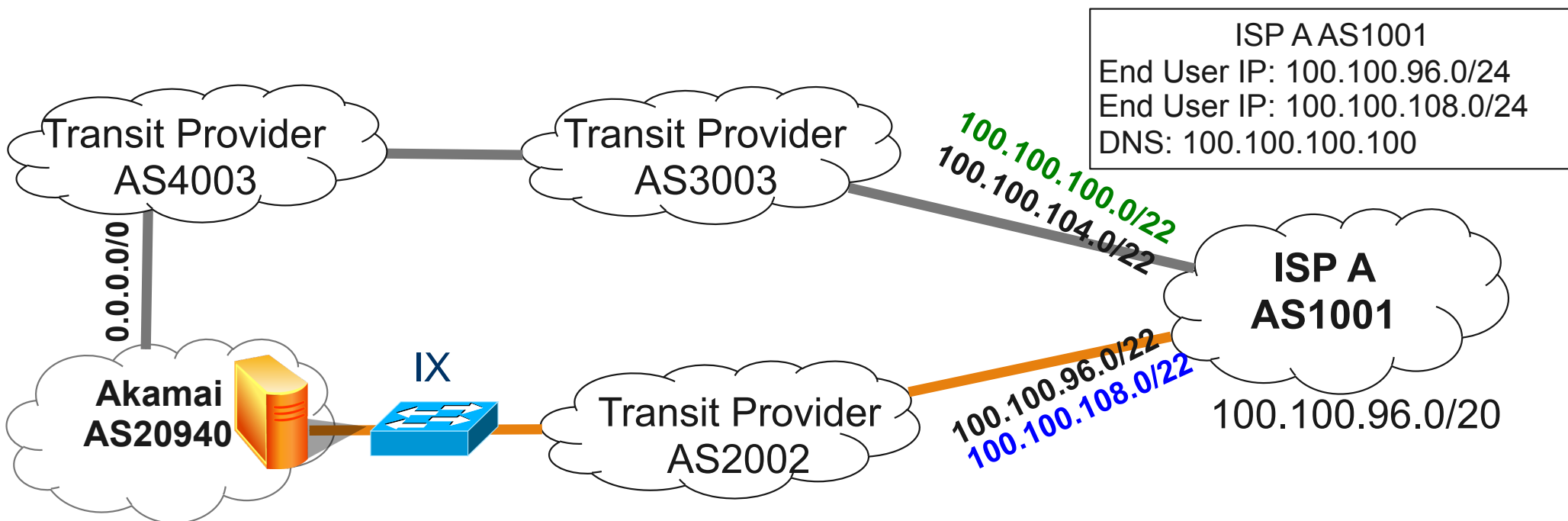
# Differing performance



- This can work perfectly fine
- But the path via the transit providers AS4003 & AS3003 might not be as good as the direct peering, [100.100.108.0/22](#) 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



100.100.96.0/22  
100.100.108.0/22



...but



24hrs later:

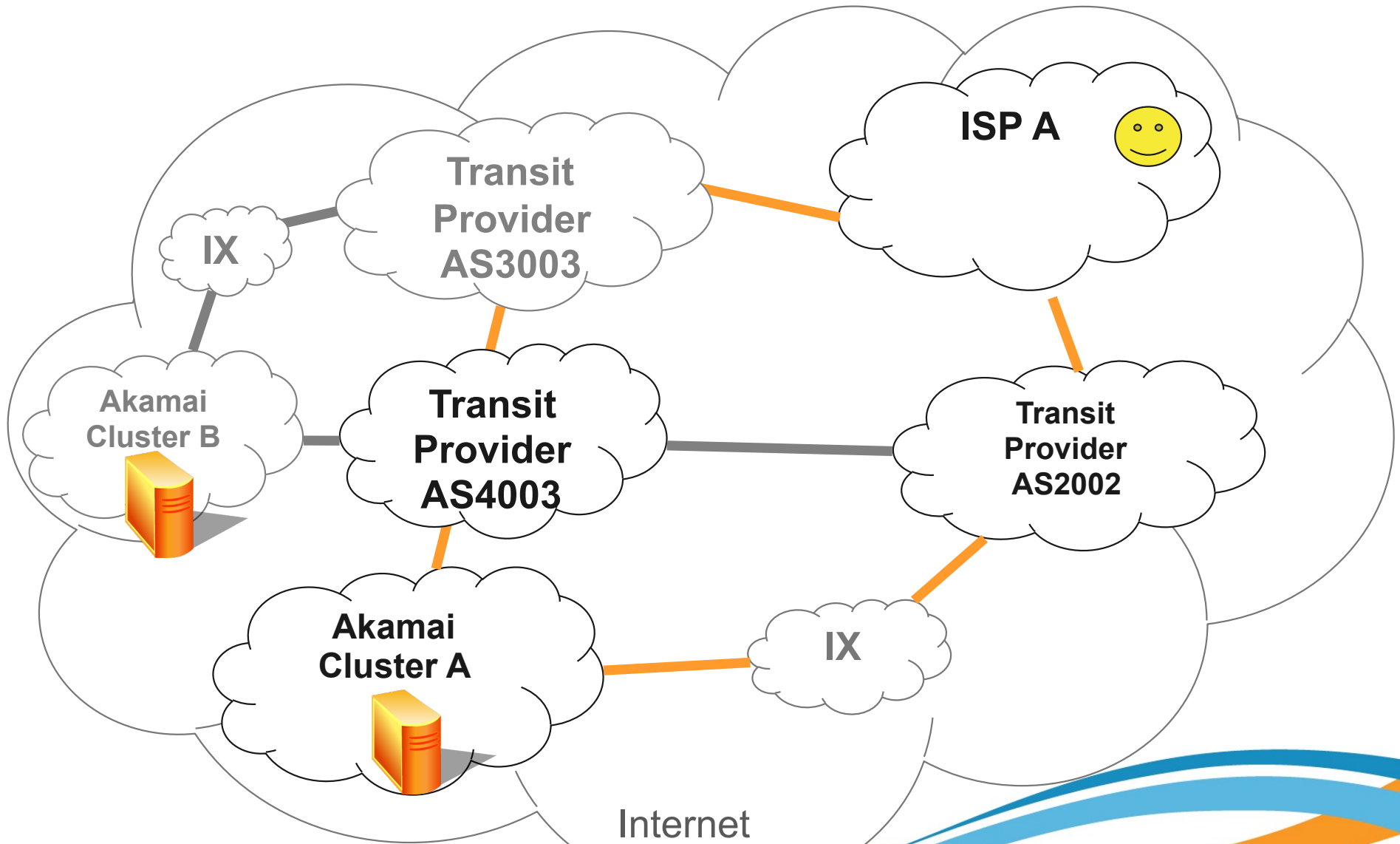
- Akamai no longer receives NS prefix **100.100.100.0/22** from AS2002
- Akamai maps the traffic of ISP A to Cluster B (where we see AS3003's prefixes) instead of Cluster A (which only peers with AS2002)
- ISP A will receive the traffic from a completely different source potentially all via AS3003 now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

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

# Before Akamai Mapping System refresh



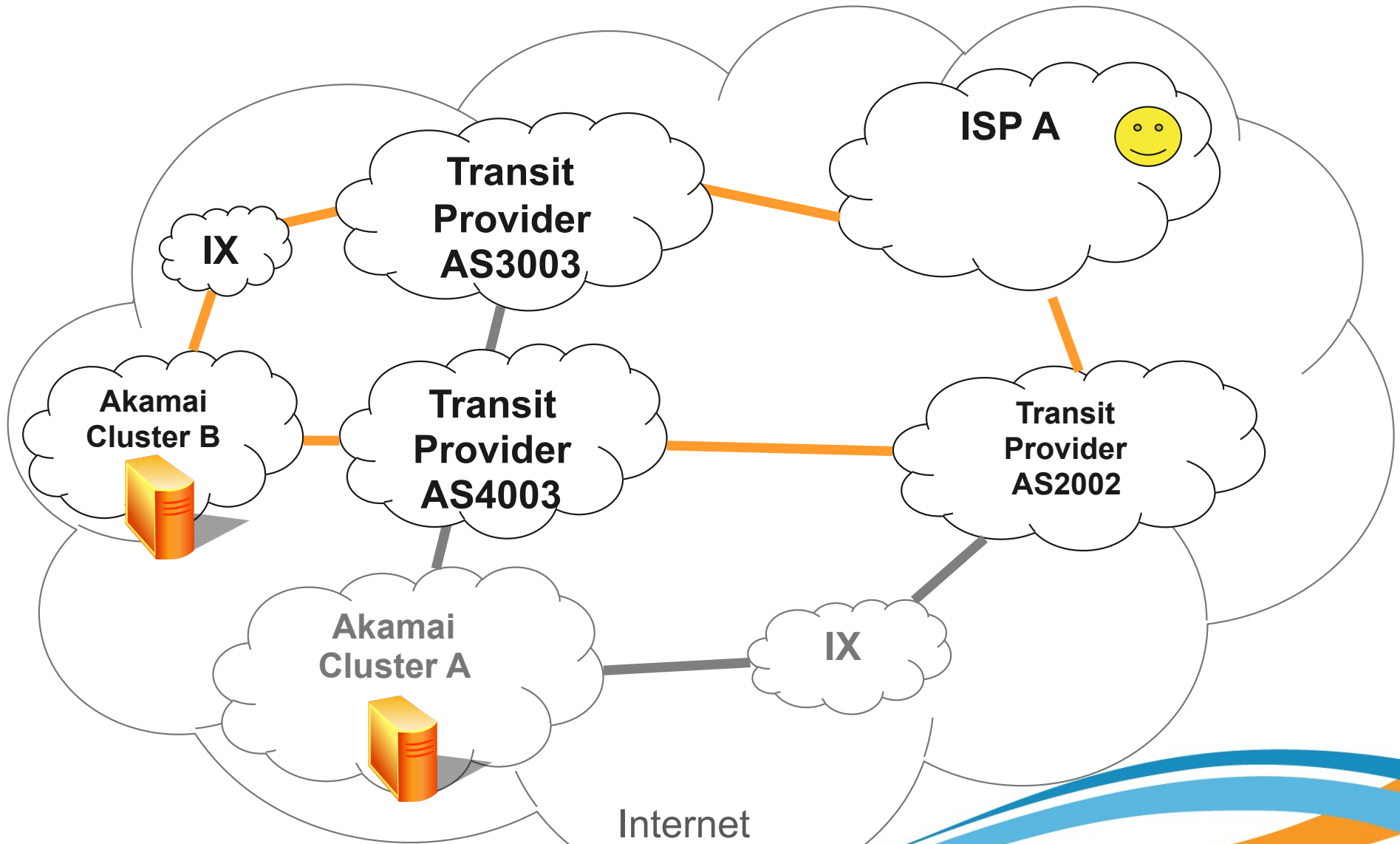
- Akamai maps the traffic to Cluster A



# After Akamai Mapping System refresh



- Akamai maps the traffic to Cluster B



## Our Recommendation

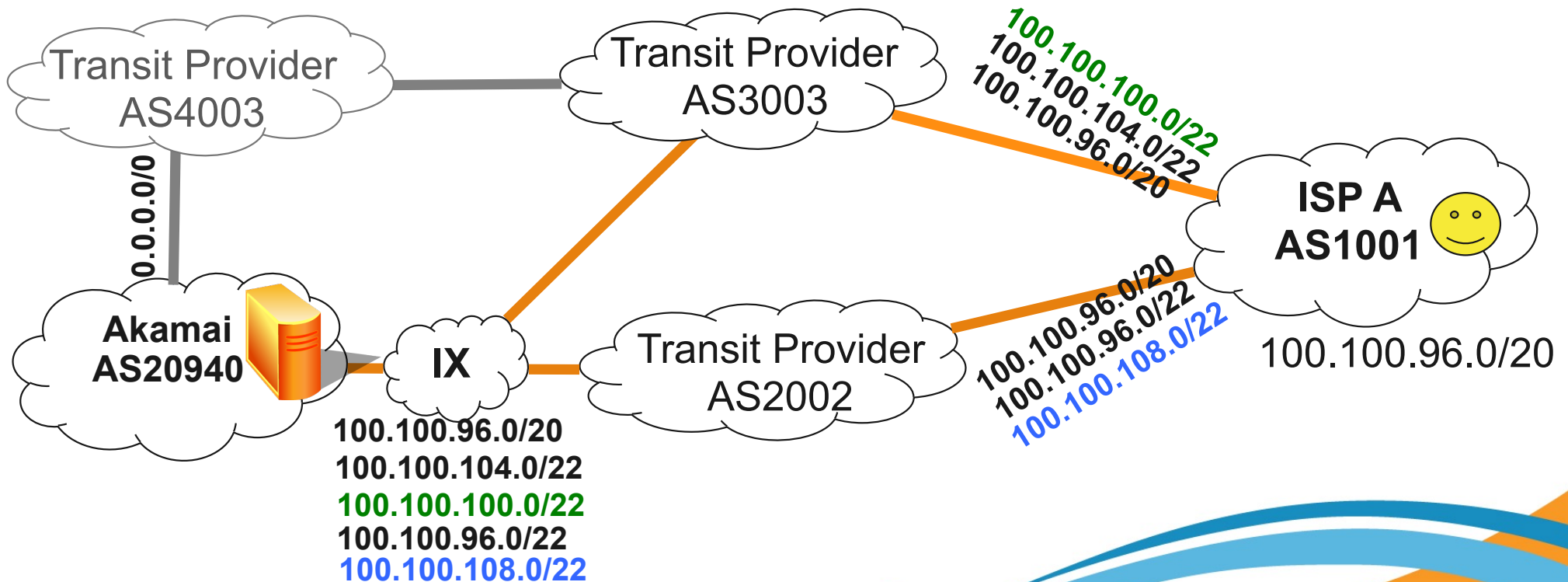


- Please maintain complete route announcement
- More specifics can be used but splitting the prefixes might have unintended effect
- Talk to us if there are any traffic or performance issues
- We can work together for traffic engineering

# Ideal solution



- ISP A should announce complete prefix in both upstream
- ISP A can work with upstream and Akamai together
- Transit Provider AS3003 can peer with Akamai





# Conclusions

Summary



# Summary



- Standard BGP traffic engineering will not have the expected results
- Changes in announcements have a delayed effect
- Mapping is based on resolving name server, 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 us for fine-tuning traffic

# Questions?



Owen DeLong [peering@akamai.com](mailto:peering@akamai.com)

More information:

Peering: <http://as20940.peeringdb.com>

<http://as32787.peeringdb.com>

Akamai 60sec: <http://www.akamai.com/60seconds>

Acknowledgements: Caglar Dabanoglu, Akamai