Peering at the Content Layer

Bruce Davie
Chief Service Provider Architect
bsd@nicira.com
Outline

• Some history: “peering peer-to-peer”
• Motivation for interconnecting CDNs
• Technical challenges
• Current efforts
• Discussion
Peering Peer-to-Peer

- See Walfish et al., IPTPS 2005
- Each DHT exposes a (put, get) service a la OpenDHT
- Service Providers operate individual DHTs and collaborate to provide a global service
Peering Providers above IP Layer

• Peering among DHTs (rather than one big DHT) provides
  – 1. Isolation: autonomous providers can manage, troubleshoot etc without concern for state in another provider
  – 2. Good Incentives: in a large ring, each provider has incentive to free-load on the resources of other providers
  – 3. Flexible Policy: e.g. a regional provider prefers to get content from local region rather than from another region

• The important thing is the interface (put, get), not the internal implementation
  – Analogous to BGP at the IP peering interface
Peering among CDN operators

- DHT-as-a-service yet to take off, but CDNs have
- So let’s peer at the Content Layer
- Peering at the IP layer provided a lot of benefits
  - Essentially took networking from the “walled garden” model of AOL (and even earlier companies) to the innovative, all-powerful, Global Internet
  - Hope for similar benefits if we can peer at Content layer
How CDNs work in one slide

• Back-end or Origin servers hold the original content
• Cache nodes or surrogates hold copies of the content, replicated and geographically distributed
• Requests for content from users get redirected to surrogates
  – Either DNS, HTTP redirect, or both
  – Lots of cleverness to pick a surrogate to optimize QOE, cost, performance, etc
• Surrogate may need to acquire content from origin or another surrogate
CDNs Today
Content delivered to user by “best” downstream CDN. Where “best” reflects Content Provider and CDN Provider policies, agreements, tariffs,..

Content Provider only need contractual/technical relationship with 1 (or a few) CDN.
Why CDN Peering Makes Sense

• There is a handful of global CDNs today, and high barrier to entry
  – Akamai, Limelight, Level3
  – They have little incentive to interconnect with anyone
• May ISPs are deploying CDN technology within their footprint because their business is increasingly about content delivery
  – CDN improves user experience by allocating resources to popular content
  – CDN saves on bandwidth costs for SP
• A lack of CDN peering means that the CDN infrastructure is underutilized
  – Netflix over Comcast example
  – ISPs lose control over how content flows across their network
  – Bandwidth wastage for ISPs
• More global-reach CDNs will be enabled by peering, fostering competition, innovation
• Akamai has been described as a money-routing company – more of that must be good, right?
CDNI Problem Overview

• Unlike DHTs, CDNs lack a really clean interface
  – Partly due to the fact that they try to be “transparent”
  – Some mess with DNS, some use HTTP-redirects

• Need to get a number of details right:
  – A request for content should produce the right content (naming)
  – A chain of CDNs deliver content – everyone wants to get paid (accounting)
  – The Content owner pays someone to deliver content - wants to know that content actually was delivered (logging)
  – Content sometimes needs to be purged in a hurry (control)
  – Lots of metadata (e.g. access controls, regional availability, time windows)
  – Routing of requests has many similarities to BGP (policy, loop-avoidance)
IETF CDNI Model and Interfaces

<== ifces in scope of CDNI
**** ifces outside scope
..... ifces outside scope

---

Upstream CDN — Control Interface — Downstream CDN

Control Interface — Logging Interface — Request Routing Interface — CDNI Metadata Interface — Acquisition Interface

Surrogate — Distribution Interface — User Agent

Request — Delivery

.................Request................. — User — Request...

Agent

+-----+

...
Example
End-User  Operator B  Operator A

| DNS  cdn.csp.com | | |
|-------------------------->|
|                          |
|                          | (1) |
| CNAME b.cdn.csp.com |
| NS records for b.cdn.csp.com |
|<------------------------|
| DNS  b.cdn.csp.com  |
|-------------------------->|
|                          |
|                          | (2) |
| IP addr of B's Delivery Node |
|<------------------------|
| HTTP  cdn.csp.com  |
|-------------------------->|
|                          |
|                          | (3) |
| DNS  op-b-acq.op-a.net |
|-------------------------->|
|                          |
|                          | (4) |
| IP addr of A's Delivery Node |
|<------------------------|
| HTTP  op-b-acq.op-a.net  |
|-------------------------->|
|                          |
|                          | (5) |
| Data |
|<------------------------|
| Data |
<------------------------|
Current Work

• Customer Trials
  – Some of these ideas tried last year with 2 operators, 2 CDN vendors, 2 countries
  – At least one larger trial underway

• IETF
  – CDNI Working Group is attempting to standardize the appropriate interfaces
  – Quite a lot of moving parts
  – Reducing the amount of ad hoc agreements and manual config seems necessary
Routing for Content

- There have been proposals to carry content names (e.g. URIs) in BGP
  - Hard to see how this can scale
  - Not obviously needed for CDNI
- Intra-CDN routing today is all proprietary
- One approach to inter-CDN routing:
  - Let CDNs advertise “footprint”
  - Redirect requests to CDNs based on matching user requests to footprint info
Discussion

• Peering at the IP layer has been key to Internet’s success
• Failure to peer at content layer is causing problems already as content dominates
• Could think of this as a cheap path to Content Centric Networking
• Akamai’s leading innovation, arguably, was routing money across peering boundaries – CDNI creates more money flow opportunities
• Global CDNs may or may not have correct incentives to play
• A lot of this will come down to economics
• Technical hurdles mostly around defining clean interfaces that meet rich set of requirements, allow flexibility in implementation
References