Odin: Microsoft’s Scalable Fault-Tolerant CDN Measurement System

Matt Calder
Manuel Schröder, Ryan Gao, Ryan Stewart, Jitendra Padhye, Ratul Mahajan, Ganesh Ananthanarayanan, Ethan Katz-Bassett
NSDI, April 2018
CDN Overview

Want low latency service for all clients

Globally distributed front-ends serve CDN traffic
Problem: What is the impact of change in a CDN?

Microsoft News Feed Application

Globally distributed front-ends serve CDN traffic

News Feed app needs IPv6

What is the global performance impact?

No idea! IPv4 and IPv6 routing is often different
Problem: What is the impact of change in a CDN?

Microsoft News Feed Application

Globally distributed front-ends

Need to understand performance impact of CDN change

Difficult to evaluate on a global scale

News Feed app wants IPv6

What is the global performance impact?

No idea! IPv4 and IPv6 routing is often different

2620:1ec:c11::/48

20 ms

420 ms

15 ms

20 ms

100 ms

Globally distributed front-ends

Global routing?
Large-scale what-if scenarios

Odin can evaluate these types of changes easily

1. Safely – don’t touch production traffic
2. Coverage of Microsoft users
3. Understand impact before deployment
1. Introduction
2. CDN Operations
3. Existing Solutions
4. Odin Design
5. CDN Operations with Odin
6. Conclusion
Internet is diverse and dynamic

• Tens of thousands of ISPs
• Factors outside of direct CDN control
  • Congestion
  • Routing changes
  • Outages
Internet is diverse and dynamic

- Tens of thousands of ISPs
- Factors outside of direct CDN control
  - Congestion
  - Routing changes
  - Outages

CDN network operations require constant insight into user performance

Internet measurement can provide this insight
Operational need for measurement

• When services become unavailable or slow, want to alert and reroute quickly

• Need diagnostic capabilities to find the root cause of issues
  • Comcast in Seattle is having trouble reaching my CDN. Are they able to reach other networks?

• Want to measure impact of changes on end-users
  • Want to take a front-end offline for maintenance. What is the performance impact on that front-end’s users?
1. Introduction
2. CDN Operations
3. Existing Solutions
4. Odin Design
5. CDN Operations with Odin
6. Conclusion
Layer 3 measurements from CDN infrastructure

• Traceroute, Ping with ICMP
• Launch from Front-ends to Internet destinations
• Issue 1: Poor coverage
  • 74% of end-users unresponsive
• Issue 2: Missing layer 7 behaviors
  • HTTP redirection
  • SSL/TLS
# Layer 3 measurements from CDN infrastructure

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Layer 3 Measurement from CDN</th>
<th>Requirements</th>
<th>Layer 3 Measurement from CDN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coverage of Microsoft Users</td>
<td>✗</td>
<td>Application Layer Measurements</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Server-side Instrumentation

• Client connections are instrumented at servers
  • Collect TCP and application layer metrics
  • Very useful but alone insufficient
• Issue 1: No explicit outage signal
• Issue 2: Alternate path exploration adds risk
# Server-side Instrumentation

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Layer 3 Measurement from CDN</th>
<th>Server-side Instrumentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coverage of Microsoft Users</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Application Layer Measurements</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Explicit Outage Signal</td>
<td>✓</td>
<td>✗</td>
</tr>
</tbody>
</table>

- Layer 3 Measurement from CDN: ✓ (present), ✗ (not present)
- Server-side Instrumentation: ✓ (present), ✗ (not present)
Third-party measurement platforms

• Operate set of vantage points on the Internet
• Run measurements on customer’s behalf
• Examples: ThousandEyes, Catchpoint, Cedexis, Dynatrace-Keynote, ...

All have inadequate coverage:
Fewer than 10 measurements per day from 88% of Microsoft customer networks
### Third-party measurement platforms

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Layer 3 Measurement from CDN</th>
<th>Server-side Instrumentation</th>
<th>Third-party Measurement Platforms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coverage of Microsoft Users</td>
<td>❌</td>
<td>✔</td>
<td>❌</td>
</tr>
<tr>
<td>Application Layer Measurements</td>
<td>❌</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Explicit Outage Signal</td>
<td>✔</td>
<td>❌</td>
<td>✔</td>
</tr>
</tbody>
</table>
Existing solutions don’t meet our needs

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Layer 3 Measurement from CDN</th>
<th>Server-side Instrumentation</th>
<th>Third-Party Measurement Platforms</th>
<th>Odin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coverage of Microsoft Users</td>
<td>✗</td>
<td>✅</td>
<td>✗</td>
<td>✅</td>
</tr>
<tr>
<td>Application Layer Measurements</td>
<td>✗</td>
<td>✅</td>
<td>✅</td>
<td>✅</td>
</tr>
<tr>
<td>Explicit Outage Signal</td>
<td>✅</td>
<td>✗</td>
<td>✅</td>
<td>✅</td>
</tr>
<tr>
<td>… (see paper for others)</td>
<td></td>
<td></td>
<td></td>
<td>✅</td>
</tr>
</tbody>
</table>
Outline

1. Introduction
2. CDN Operations
3. Existing Solutions
4. Odin Design
5. CDN Operations with Odin
6. Conclusion
Odin Design

1. Client-side Platform
2. Active Measurement
3. Application Layer
1. Client-side Platform
2. Active Measurement
3. Application Layer
4. Both Web and Rich Clients
Odin Design

1. Client-side Platform
2. Active Measurement
3. Application Layer
4. Both Web and Rich Clients
5. Explicit Failure Notification
Examples showed measurements to the application server

Want richer measurements
Odin Design

Orchestration Service

Offline Analysis

Online Alerting

Microsoft

Primary and backup report endpoints

Target URLs

"MeasurementEndpoints": [
  {"endpoint": "m1.contoso.com"},
  {"endpoint": "m2.microsoft.com"},
  {"endpoint": "m3.azure.com"}],

"ReportEndpoints": ["r1.azure.com", "r2.othercdn.com"]
Odin Design

Microsoft U.S.

Server B

Orchestration Service

Offline Analysis

Online Alerting

Report Endpoint

Microsoft Europe

m1.contoso.com: 20ms

GET tiny.png

m3.contoso.com
Odin Design: Fault tolerance

Need to receive measurements even if Microsoft’s network is unavailable
Odin Design: Fault tolerance

Server B

Orchestration Service

Report Endpoint

Microsoft

Offline Analysis

Online Alerting

GET tiny.gif

Odin

3rd Party Network

Report Proxy

B: ERROR
Paper discusses cases where failures impact primary and proxy report endpoints.
Outline

1. Introduction
2. CDN Operations
3. Existing Solutions
4. Odin Design
5. CDN Operations with Odin
6. Conclusion
Supporting Operations: Odin’s DNS redirection maps

Azure Global Application Traffic Management

- Azure - 38 Cloud Regions
- Customers deploy services in a subset for resilience & perf
- Cloud apps load balanced using DNS
- Direct user to lowest-latency region
Supporting Operations: Odin’s DNS redirection maps

Azure Global Application Traffic Management

Azure - 38 Cloud Regions

Services in a subset for resilience & perf

Cloud apps load balanced using DNS

Direct user to lowest-latency region

Odin can find the lowest latency region for users
DNS-based redirection for CDNs

Connect to Region A

2.2.2.2

IP for www.cdn.com?

Local DNS (LDNS)
1.2.3.4

IP for www.cdn.com?

Region A

2.2.2.2

Region B

3.3.3.3

Authoritative DNS for cdn.com
Previous Approach: Predicting best region for LDNS

Ping all LDNSes from all Regions

Direct LDNS to Region with minimum ping

Limitations:
- Most LDNS don’t respond to ping
- Ping to LDNS may not reflect client performance

LDNS 4.4.4.4

Region A

10ms

LDNS 5.5.5.5

Region B

20ms

LDNS 6.6.6.6

Region C

80ms
Odin’s Approach: Predicting best region for LDNS

- Odin knows client => LDNS
- Know all clients served by an LDNS
- Associate client to region measurements with client’s LDNS
- Compare median latency toward each region
- Select lowest latency region as best
Improving cloud performance with Odin

Japan, Italy, Spain over 25% latency improvement

10 countries see > 10% improvement in latency
Supporting Operations: Using Odin to identify outages

- Widespread Internet availability issues from AS1759?
- Total outage between AS1759 & Microsoft?
- Is Helsinki front-end failing?
- Regional outage between AS1759 & Microsoft in Helsinki?
Supporting Operations: Using Odin to identify outages

- Widespread Internet availability issues from AS1759?
- Total outage between AS1759 & Microsoft?
- Is Helsinki front-end failing?
- Regional outage between AS1759 & Microsoft in Helsinki?
Supporting Operations: Using Odin to identify outages

- Widespread Internet availability issues from AS1759?
- Total outage between AS1759 & Microsoft?
- Is Helsinki front-end failing?
- Regional outage between AS1759 & Microsoft in Helsinki?
Supporting Operations: Using Odin to identify outages

- Widespread Internet availability issues from AS1759?
- Total outage between AS1759 & Microsoft?
- Is Helsinki front-end failing?
- Regional outage between AS1759 & Microsoft in Helsinki?
Conclusion

• **Odin** is a client-side, active measurement platform to support Microsoft CDN operations
  • Running in production for over 2 years
  • Deployed in key Microsoft end-user applications
  • Billions of measurements per day

• Overcomes limitations of existing approaches

• Improves CDN operations
  • Detect and diagnose outages
  • Odin-based DNS-redirection for cloud customers