

IPv6 Client brokenness

auch IPv6 brokenness oder Dual-Stack Problem

Ulrich Hauser, Network Engineer, Ifolor AG

- Warum gibt es das Problem?
- Überlegungen und Vorbereitungen zur Erfassung / Messung
- Wie erfolgt die Erfassung / Messung?
- Erste Erkenntnisse
- Optimierung der Erfassung / Messung
- Übersicht der Ergebnisse
- Detaillierter Blick auf einzelne Clientverbindungen
- Ergebnisse nach manueller Kontrolle
- Das würden wir beim nächsten Mal anders machen
- Fragen

- Wenn für das Zielsystem (Web-Server) publiziert wird, dass es über IPv4 und IPv6 erreichbar ist (A und AAAA-DNS-Record), sollte IPv6 als neues Protokoll bevorzugt genutzt werden.
- Einige Systeme meinen IPv6 nutzen zu können. Der Weg ins globale IPv6-Internet läuft ins leere; ohne auf die Gründe einzugehen.
- Client Brokenness ist seit vielen Jahren ein Thema, nicht nur bei Update-Resistenten Anwendern
- Teilweise wurde mit einem Workaround gestartet
 - www.six.heise.de (März 2009)
 - ipv6.google.com
- Heise IPv6 Day am 16.Sept.2010, seit 29.Sept.2010 mit IPv4 und IPv6 online (J.Endress: „Wir haben technisch interessierte Kunden.“)
- Messungen von Tore Anderson in Norwegen in 2010/2011: < 0,025%
- Grund für den World IPv6 (Test) Day (8.Juni 2011) und den World IPv6 Launch Day (6.Juni 2012) der ISCO mit Unterstützung von Facebook, Google, Yahoo, Akamai und vielen anderen

Darf es Client Brokenness noch geben?

- Darf es Client Brokenness überhaupt noch geben wenn Google, Facebook, YouTube und viele andere DualStack-DNS-Records publizieren?

```
> nslookup -querytype=A+AAAA www.google.com
```

```
Nicht autorisierende Antwort:
```

```
Name:          www.google.com
```

```
Addresses:    2a00:1450:400f:800::1014
```

```
              173.194.32.20
```

```
              173.194.32.17
```

- **leider JA (es gibt noch andere Anbieter):**

```
> nslookup -querytype=A+AAAA www.bing.com
```

```
Nicht autorisierende Antwort:
```

```
Name:         any.edge.bing.com
```

```
Address:      204.79.197.200
```

```
Aliases:      www.bing.com
```

```
> nslookup -querytype=A+AAAA www.linkedin.com
```

```
Nicht autorisierende Antwort:
```

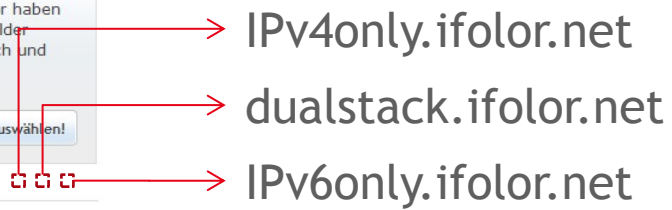
```
Name:         www.linkedin.com
```

```
Address:      91.225.248.80
```

Vorbereitung: Statistik Version 1 der Clientzugriffe (1)



- Web-Seite bleibt für diesen Test weiterhin nur über IPv4 erreichbar -> keine Kundenimpact
- Auf der www-Seite werden drei kleine, transparente Bildchen eingebaut, die von drei unterschiedlichen www-Servern abgerufen werden
- Wenn der Client IPv6 kann, soll IPv6 als Protokoll verwendet werden
- Kein Tracking-Cookie



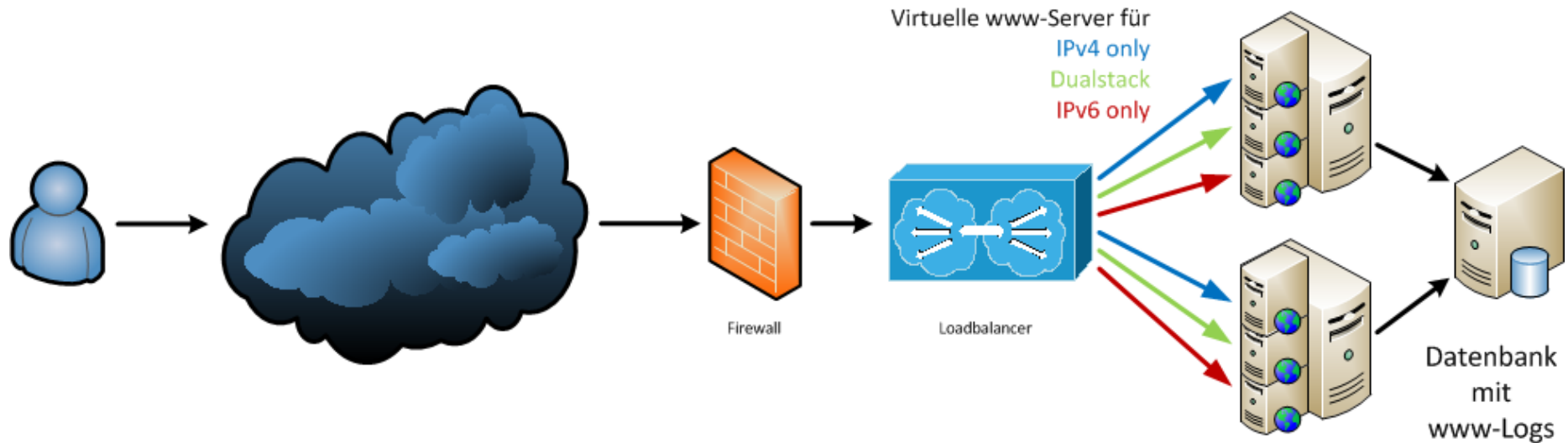
Messmethode von Tore Anderson (www.fud.no/ipv6) ähnlich Frank Mayer, (iX 7/2011)

- Alle Log-Dateien der Zähl-Server zusammenfügen
- Basierend auf Datum, Zeit, User-Agent und Referer wird von Logging-System eine Request-ID erzeugt; ohne die IP-Adresse (IPv4 != IPv6).
- Wie viele Logeinträge = Logzeilen gibt es je Request-ID?

Logzeilen je Request-ID	Was kann der Client	Zu erwartender Impact
Zwei Logzeilen (2 x IPv4-Adresse)	Nur IPv4	Keiner
Drei Logzeilen (IPv4 und IPv6-Adresse)	IPv4 und IPv6 funktionieren	Keiner
Eine Logzeile (nur IPv4only Bild)	Client meint IPv6 zu können, IPv6 kommt nicht zum Zählserver	Falls vorhanden: System hat Client Brokenness Problem
Zwei Logzeilen (2 x IPv6-Adresse)	Nur IPv6	Wie kann der Kunde die www-Seite ohne AAAA- Record aufrufen?

Anmerkungen zur Konfiguration

- DNS-TTL mit 10 Sekunden
- Kein Keepalive
- No cache allowed



nur Requests zählen ist zu wenig...

- Auf dem Client:
img src=http://ipv4only.ifolor.net/tracking1x1.gif
img src=http://dualstack.ifolor.net/tracking1x1.gif
img src=http://ipv6only.ifolor.net/tracking1x1.gif

Ergebnis:

Wir haben an manchen Tagen mehr Dual-Stack Requests als IPv4 only requests gesehen. ☹️

Diesen Effekt gibt es nicht nur bei uns, wie am Beispiel von Eric Vyncke <http://vyncke.org/testv6/> zu sehen ist:

Date		IPv4-only	Dual-stack	IPv6-only	IPv6 broken
Since last month	2014-01-25	7367	7373	2749	-0.02 %
Since last week	2014-02-18	1903	1905	742	-0.04 %
Since yesterday	2014-02-24	414	410	115	0.46 %

- Um eine zuverlässige Zuordnung der unterschiedlichen Requests zu einem Client zu erreichen wird eine möglichst eindeutige Request-ID benötigt.
- Es soll eine Zufallszahl erzeugt werden. Diese soll als eindeutige Request-ID die Zuordnung der Log-Zeilen vom http-Server zu einem Client ermöglichen; unabhängig von der IP-Version.
Diese Request-ID dient als Basis für die Statistik.
- Mit ein bisschen Cut-and-Paste von einem vorhandene Javascript wurde die Zählung auf den folgenden Zeilen umgestellt:

```
<script language="javascript" type="text/javascript">
rdm=Math.random()*10000000+10000000;
document.write('');
document.write('');
document.write('');
</script>
```

Falls der Browser kein JavaScript unterstützt oder JavaScript deaktiviert müssen wir diese Requests besonders behandeln.

Natürlich würde es auch mit PHP gehen. Dann generiert der Server die Request-ID:

```
<?php
$rdm = mt_rand(1,mt_getrandmax());
// echo "$rdm <br> \n";
echo "<img src=\"http://ipv4only.ifolor.net/tracking1x1.gif?rdm=
    $rdm\" width=\"1px\" height=\"1px\" border=\"0px\" alt=\"\" >
    <br> \n";
echo "<img src=\"http://dualstack.ifolor.net/tracking1x1.gif?rdm=
    $rdm\" width=\"1px\" height=\"1px\" border=\"0px\" alt=\"\" >
    <br> \n";
echo "<img src=\"http://ipv6only.ifolor.net/tracking1x1.gif?rdm=
    $rdm\" width=\"1px\" height=\"1px\" border=\"0px\" alt=\"\" >
    <br> \n";
?>
```

- Alle Log-Dateien der Zähl-Server zusammenfügen
- ~~Basierend auf Datum, Zeit, User-Agent und Referer wird von Logging-System eine Request-ID erzeugt; ohne IP-Adresse..~~ Request-ID erzeugt jetzt das JavaScript
- Wie viele Logzeilen gibt es je Request-ID?

Logzeilen je Request-ID	Was kann der Client	Zu erwartender Impact
Zwei Logzeilen (2 x IPv4-Adresse)	Nur IPv4	Keiner
Drei Logzeilen (IPv4 und IPv6-Adresse)	IPv4 und IPv6 funktionieren	Keiner
Eine Logzeile (nur IPv4only Bild)	Client meint IPv6 zu können, Request kommt jedoch nicht zum Zählserver	Falls vorhanden: System hat Client Brokenness Problem
Zwei Logzeilen (2 x IPv6-Adresse)	Nur IPv6	Wie kann der Kunde die www-Seite ohne AAAA- Record aufrufen?
Mehr als drei Logzeilen	???	Falls es das gibt, Zeilen manuell prüfen

- Search-String im Splunk:

index="iis_count" cs_uri_stem="tracking1x1.gif" | sort by date,time | table cs_uri_query c_ip cs_host date time host

cs_uri_query ↕	c_ip ↕	cs_host ↕	date ↕	time ↕	host ↕
rdm=14678570.532819767	2	9.179	dualstack.ifolor.net	2013-11-29 19:23:05	2
rdm=14678570.532819767	2	9.179	ipv4only.ifolor.net	2013-11-29 19:23:05	1
rdm=11191464.451075464	2	9.179	ipv4only.ifolor.net	2013-11-29 19:35:21	1
rdm=11191464.451075464	2	9.179	dualstack.ifolor.net	2013-11-29 19:35:21	2
rdm=11675865.443759121	2	a:f180:4c3d:2b76:c75a:9ab2	ipv6only.ifolor.net	2013-11-29 20:16:38	2
rdm=11675865.443759121	2	a:f180:4c3d:2b76:c75a:9ab2	dualstack.ifolor.net	2013-11-29 20:16:38	1
rdm=11675865.443759121	2	9.180	ipv4only.ifolor.net	2013-11-29 20:16:38	1
rdm=19981187.728577025	2	a:f180:4c3d:2b76:c75a:9ab2	dualstack.ifolor.net	2013-11-29 21:15:12	1
rdm=19981187.728577025	2	a:f180:4c3d:2b76:c75a:9ab2	ipv6only.ifolor.net	2013-11-29 21:15:12	2
rdm=19981187.728577025	2	9.180	ipv4only.ifolor.net	2013-11-29 21:15:12	2
rdm=12506540.438625962	2	9.180	ipv4only.ifolor.net	2013-11-29 21:31:05	2
rdm=12506540.438625962	2	9.180	dualstack.ifolor.net	2013-11-29 21:31:05	1
rdm=12181397.355161607	2	a:f180:ec6a:af7c:b293:f907	ipv6only.ifolor.net	2013-11-30 07:56:04	2
rdm=12181397.355161607	2	a:f180:ec6a:af7c:b293:f907	dualstack.ifolor.net	2013-11-30 07:56:04	1
rdm=12181397.355161607	2	9.180	ipv4only.ifolor.net	2013-11-30 07:56:04	1
rdm=13159704.725258052	7	.70	ipv4only.ifolor.net	2013-12-02 09:06:54	2
rdm=13159704.725258052	7	.70	dualstack.ifolor.net	2013-12-02 09:06:54	1
rdm=18513289.101312757	7	.70	ipv4only.ifolor.net	2013-12-02 12:28:06	1

Die Idee mit Zählen der Request-ID funktioniert!



```
index=" " cs_uri_stem=" gif" | stats count by cs_uri_query | eval  
state=case(count>3,"unklar",count=3,"super",count=2,"OK",count=1,"Not OK") | sort by count
```

18 events (25.11.13 17:00:00,000 to 02.12.13 17:30:51,000)

Job Abgeschlossen

Ereignisse (18) Statistics (8) Visualisierung

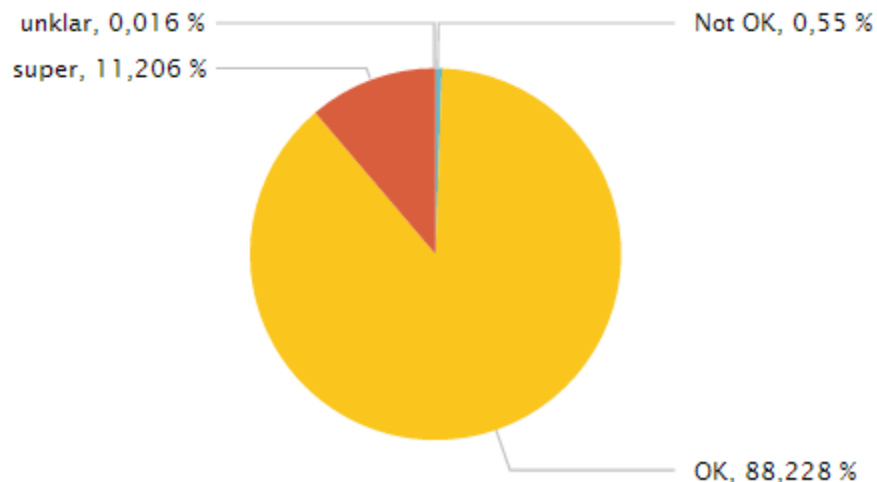
100 Per Page Format Vorschau

cs_uri_query	count	state
rdm=18513289.101312757	1	Not OK
rdm=11191464.451075464	2	OK
rdm=12506540.438625962	2	OK
rdm=13159704.725258052	2	OK
rdm=14678570.532819767	2	OK
rdm=11675865.443759121	3	super
rdm=12181397.355161607	3	super
rdm=19981187.728577025	3	super

Ende 2013 wurden die ersten Zählpixel auf wenig besuchten Seiten aktiv gesetzt, seit Januar 2014 auf den Startseiten und einigen anderen Seiten

Basierend auf über 2 Millionen Log-Zeilen sieht die **automatische** Auswertung wie folgt aus:

- (88.883%) 88.228% Request-IDs mit IPv4 only
 - (10.752%) 11.206% Request-IDs können auch IPv6 (Dual-Stack)
 - (99.635%) 99.434% KEINE Probleme mit Client-Brokenness zu erwarten
 - (0.352%) 0.550% Request-IDs mit 1 Zeile (not-OK) ☹
 - (0.013%) 0.016% Request-IDs in mehr als 3 Zeilen zu finden
- Stand März 2014 Mai 2014



- Im folgenden werden einige Beispiele gezeigt, die auffällig waren und daher genauer betrachtet worden sind
- Meistens war der Anlass eine „Not-OK“ oder „unklar“ Markierung, die anschliessend verifiziert wurde

Ist dieser Client wirklich „Not OK“?

Ein Beispiel für einen Client, der sehr sicher OK ist aber aus bisher nicht erklärbaren Gründen bei einem späteren Besuch der Web-Seite nur den dualstack-Pixel über IPv4 angefordert hat:

cs_uri_query	c_ip	cs_host	date	time	cs_User_Agent	
rdm=15087073.799222708	62	0.227	dualstack	2013-12-30	10:00:49	Mozilla/5.0+(iPhone;+CPU+iPhone+OS+7_0_4+like+Mac+OS+X
rdm=15087073.799222708	62	0.227	ipv4only	2013-12-30	10:00:49	Mozilla/5.0+(iPhone;+CPU+iPhone+OS+7_0_4+like+Mac+OS+X
rdm=13942959.350533783	62	0.227	dualstack	2014-01-01	15:11:09	Mozilla/5.0+(iPhone;+CPU+iPhone+OS+7_0_4+like+Mac+OS+X

Zur Erinnerung: Statistik von Eric Vyncke

Date	IPv4-only	Dual-stack	IPv6-only	IPv6 broken
Since last month	7367	7373	2749	-0.02 %
Since last week	1903	1905	742	-0.04 %
Since yesterday	414	410	115	0.46 %

Ist dieser Client wirklich „Not OK“?

Ein Beispiel für einen Client, der sehr sicher mit DualStack Web-Seiten kein Problem hat aber bei unserer Erfassung ein überraschendes Verhalten zeigt:

cs_uri_query	c_ip	cs_host	date	time	cs_User_Agent_
-	15	3.98 dualstack	2014-02-05	09:32:37	Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT+6.1;+WOW64;+Trident/4.0;+SL
rdm=14558585.007543023	15	3.98 dualstack	2014-02-05	09:32:37	Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT+6.1;+WOW64;+Trident/4.0;+SL
-	15	3.98 ipv4only	2014-02-05	09:32:37	Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT+6.1;+WOW64;+Trident/4.0;+SL
rdm=14558585.007543023	15	3.98 ipv4only	2014-02-05	09:32:37	Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT+6.1;+WOW64;+Trident/4.0;+SL
rdm=10058790.153897481	15	3.98 dualstack	2014-02-05	09:36:35	Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT+6.1;+WOW64;+Trident/4.0;+SL
rdm=14457535.055990008	15	3.98 ipv4only	2014-02-05	09:36:55	Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT+6.1;+WOW64;+Trident/4.0;+SL

Ist dieser Client wirklich „Not OK“?

Ein Beispiel für einen Client, der sehr sicher OK ist aber aus bisher nicht erklärbaren Gründen bei einem Besuch der Web-Seite nicht immer beide Pixel angefordert hat:

cs_uri_query	c_ip	cs_host	date	time	cs_User_Agent_
rdm=19865390.68810922	217	8.125	dualstack	2014-02-10 21:33:32	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10.9;+rv:26.0)+Geck
rdm=19865390.68810922	217	8.125	ipv4only	2014-02-10 21:33:32	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10.9;+rv:26.0)+Geck
rdm=10004809.213559888	217	8.125	ipv4only	2014-02-13 23:38:54	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10.9;+rv:26.0)+Geck
rdm=11895626.018580303	217	8.125	dualstack	2014-02-19 22:04:19	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10.9;+rv:27.0)+Geck
rdm=18574905.57785186	217	8.125	dualstack	2014-03-02 19:05:22	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10.9;+rv:27.0)+Geck
rdm=18574905.57785186	217	8.125	ipv4only	2014-03-02 19:05:22	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10.9;+rv:27.0)+Geck

Ist dieser Client wirklich „Not OK“?

Ein Beispiel für einen Client, der sehr sicher OK ist aber aus bisher nicht erklärbaren Gründen bei einem Request keinen Referer (von welcher Web-Seite wurde ich aufgerufen) gesendet hat:

cs_uri_query	c_ip	cs_host	date	time	cs_Referer_	cs_User_Agent_
rdm=10003610.561320884	82	1.205	dualstack	2014-01-12 19:49:21	http://www.ifolor.ch/en	Mozilla/5.0+(compatible;+MSIE+9.0;+Windows+NT+6.1;+...
rdm=10003610.561320884	82	1.205	ipv4only	2014-01-12 19:49:23	-	Mozilla/5.0+(compatible;+MSIE+9.0;+Windows+NT+6.1;+...

Ist dieser Client wirklich „Not OK“?



Ein Beispiel für einen Client, der sehr sicher OK ist aber aus bisher nicht geklärten Gründen nicht immer alle Pixel angefordert hat:

cs_uri_query	c_ip	cs_host	date	time	cs_User_Agent
rdm=1467150807	77	.123	dualstack	2014-02-20 20:27:22	Mozilla/5.0+(iPhone;+CPU+iPhone+OS+6_1_3+like+Mac+OS+X)+AppleWebKit/536.26+(KHTML, like Gecko) Chrome/30.0.1599.101 Mobile Safari/536.26
rdm=1467150807	77	.123	ipv4only	2014-02-20 20:27:22	Mozilla/5.0+(iPhone;+CPU+iPhone+OS+6_1_3+like+Mac+OS+X)+AppleWebKit/536.26+(KHTML, like Gecko) Chrome/30.0.1599.101 Mobile Safari/536.26
rdm=722362840	77	.123	ipv4only	2014-02-20 20:28:26	Mozilla/5.0+(iPhone;+CPU+iPhone+OS+6_1_3+like+Mac+OS+X)+AppleWebKit/536.26+(KHTML, like Gecko) Chrome/30.0.1599.101 Mobile Safari/536.26
rdm=1931103769	77	.123	dualstack	2014-02-20 20:28:35	Mozilla/5.0+(iPhone;+CPU+iPhone+OS+6_1_3+like+Mac+OS+X)+AppleWebKit/536.26+(KHTML, like Gecko) Chrome/30.0.1599.101 Mobile Safari/536.26
rdm=1931103769	77	.123	ipv4only	2014-02-20 20:28:35	Mozilla/5.0+(iPhone;+CPU+iPhone+OS+6_1_3+like+Mac+OS+X)+AppleWebKit/536.26+(KHTML, like Gecko) Chrome/30.0.1599.101 Mobile Safari/536.26
rdm=1959733521	77	.123	dualstack	2014-02-23 18:21:31	Mozilla/5.0+(iPhone;+CPU+iPhone+OS+7_0_4+like+Mac+OS+X)+AppleWebKit/537.51.1+(KHTML, like Gecko) Chrome/30.0.1599.101 Mobile Safari/537.51.1
rdm=1959733521	77	.123	ipv4only	2014-02-23 18:21:31	Mozilla/5.0+(iPhone;+CPU+iPhone+OS+7_0_4+like+Mac+OS+X)+AppleWebKit/537.51.1+(KHTML, like Gecko) Chrome/30.0.1599.101 Mobile Safari/537.51.1
rdm=1137584686	77	.123	dualstack	2014-02-23 18:24:25	Mozilla/5.0+(iPhone;+CPU+iPhone+OS+7_0_4+like+Mac+OS+X)+AppleWebKit/537.51.1+(KHTML, like Gecko) Chrome/30.0.1599.101 Mobile Safari/537.51.1
rdm=1137584686	77	.123	ipv4only	2014-02-23 18:24:25	Mozilla/5.0+(iPhone;+CPU+iPhone+OS+7_0_4+like+Mac+OS+X)+AppleWebKit/537.51.1+(KHTML, like Gecko) Chrome/30.0.1599.101 Mobile Safari/537.51.1
rdm=1476841668	77	.123	dualstack	2014-02-23 18:25:30	Mozilla/5.0+(iPhone;+CPU+iPhone+OS+7_0_4+like+Mac+OS+X)+AppleWebKit/537.51.1+(KHTML, like Gecko) Chrome/30.0.1599.101 Mobile Safari/537.51.1
rdm=1476841668	77	.123	ipv4only	2014-02-23 18:25:30	Mozilla/5.0+(iPhone;+CPU+iPhone+OS+7_0_4+like+Mac+OS+X)+AppleWebKit/537.51.1+(KHTML, like Gecko) Chrome/30.0.1599.101 Mobile Safari/537.51.1
rdm=618526890	77	.123	dualstack	2014-02-23 18:25:41	Mozilla/5.0+(iPhone;+CPU+iPhone+OS+7_0_4+like+Mac+OS+X)+AppleWebKit/537.51.1+(KHTML, like Gecko) Chrome/30.0.1599.101 Mobile Safari/537.51.1
rdm=618526890	77	.123	ipv4only	2014-02-23 18:25:41	Mozilla/5.0+(iPhone;+CPU+iPhone+OS+7_0_4+like+Mac+OS+X)+AppleWebKit/537.51.1+(KHTML, like Gecko) Chrome/30.0.1599.101 Mobile Safari/537.51.1
rdm=1238369843	77	.123	dualstack	2014-02-23 18:25:52	Mozilla/5.0+(iPhone;+CPU+iPhone+OS+7_0_4+like+Mac+OS+X)+AppleWebKit/537.51.1+(KHTML, like Gecko) Chrome/30.0.1599.101 Mobile Safari/537.51.1
rdm=1238369843	77	.123	ipv4only	2014-02-23 18:25:52	Mozilla/5.0+(iPhone;+CPU+iPhone+OS+7_0_4+like+Mac+OS+X)+AppleWebKit/537.51.1+(KHTML, like Gecko) Chrome/30.0.1599.101 Mobile Safari/537.51.1

Was ist mit diesem Dual-Stack Client?

- Viele korrekte Requests
- Für Dualstack-Ziel teilweise IPv4 und teilweise IPv6
- Warum ein paar Tage später nur ein Request? (dann aber mit IPv6 ☺)
- Offtopic: Privacy funktioniert bei dem Client☺

cs_uri_query	c_ip	cs_host	date	time	cs_User_Agent
rdm=17302939.45176527	17	dualstack	2014-01-18	17:06:07	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_7_4)+AppleWebKit/534.57.5+
rdm=17302939.45176527	2a	:db60:14f3 ipv6only	2014-01-18	17:06:07	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_7_4)+AppleWebKit/534.57.5+
rdm=17302939.45176527	17	ipv4only	2014-01-18	17:06:07	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_7_4)+AppleWebKit/534.57.5+
rdm=10187146.565876901	17	ipv4only	2014-01-18	17:07:37	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_7_4)+AppleWebKit/534.57.5+
rdm=10187146.565876901	2a	:db60:14f3 ipv6only	2014-01-18	17:07:37	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_7_4)+AppleWebKit/534.57.5+
rdm=10187146.565876901	17	dualstack	2014-01-18	17:07:38	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_7_4)+AppleWebKit/534.57.5+
rdm=19049090.524204075	2a	:db60:14f3 ipv6only	2014-01-18	17:36:45	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_7_4)+AppleWebKit/534.57.5+
rdm=19049090.524204075	17	dualstack	2014-01-18	17:36:45	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_7_4)+AppleWebKit/534.57.5+
rdm=19049090.524204075	17	ipv4only	2014-01-18	17:36:45	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_7_4)+AppleWebKit/534.57.5+
rdm=10264900.638721883	17	ipv4only	2014-01-18	17:53:32	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_7_4)+AppleWebKit/534.57.5+
rdm=10264900.638721883	2a	:db60:14f3 ipv6only	2014-01-18	17:53:32	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_7_4)+AppleWebKit/534.57.5+
rdm=10264900.638721883	17	dualstack	2014-01-18	17:53:32	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_7_4)+AppleWebKit/534.57.5+
rdm=18066862.25347221	17	ipv4only	2014-01-18	17:53:32	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_7_4)+AppleWebKit/534.57.5+
rdm=18066862.25347221	2a	:db60:14f3 ipv6only	2014-01-18	17:53:32	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_7_4)+AppleWebKit/534.57.5+
rdm=18066862.25347221	17	dualstack	2014-01-18	17:53:32	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_7_4)+AppleWebKit/534.57.5+
rdm=13169593.424536288	2a	:db60:821:dualstack	2014-01-20	10:27:55	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_7_4)+AppleWebKit/534.57.5+
rdm=13169593.424536288	2a	:db60:821:dualstack	2014-01-20	10:27:55	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_7_4)+AppleWebKit/534.57.5+
rdm=13169593.424536288	17	ipv4only	2014-01-20	10:27:55	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_7_4)+AppleWebKit/534.57.5+
rdm=10106246.150098741	2a	:db60:f15f:dualstack	2014-01-24	16:20:04	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_7_4)+AppleWebKit/534.57.5+

Kann die Request-ID doppelt vorkommen?

- Die Request-ID ist in unserem Fall eine (Pseudo-) Zufallszahl.
- Per Zufall kann diese auch von zwei Clients identisch generiert werden

cs_uri_query	c_ip	cs_host	date	time	cs_User_Agent_	
rdm=10202310.145832598	85	1.178	dualstack	2014-01-15	09:44:53	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_7_5)+AppleWebKit/537.73.11+(KHTML,+
rdm=10202310.145832598	85	1.178	ipv4only	2014-01-15	09:44:53	Mozilla/5.0+(Macintosh;+Intel+Mac+OS+X+10_7_5)+AppleWebKit/537.73.11+(KHTML,+
rdm=10202310.145832598	80	09.28	dualstack	2014-02-09	10:12:15	Mozilla/5.0+(iPad;+CPU+OS+7_0_4+like+Mac+OS+X)+AppleWebKit/537.51.1+(KHTML,+
rdm=10202310.145832598	80	09.28	ipv4only	2014-02-09	10:12:15	Mozilla/5.0+(iPad;+CPU+OS+7_0_4+like+Mac+OS+X)+AppleWebKit/537.51.1+(KHTML,+

Manchmal kommt der Dual Stack Request verzögert....



- Hat hier Happy Eyeball zugeschlagen?
(IPv6 Request fehlgeschlagen, dann probiert der Browser auch IPv4)

cs_uri_query	c_ip	cs_host	date	time	cs_User_Agent
rdm=10216331.837364519	145.2...0.254	dualstack	2014-02-06	16:54:08	Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT+6.1;+Trident/4.0;+SLC
rdm=10216331.837364519	145.2...0.254	ipv4only	2014-02-06	16:54:08	Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT+6.1;+Trident/4.0;+SLC
rdm=10216331.837364519	145.2...0.254	ipv4only	2014-02-06	16:54:20	Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT+6.1;+Trident/4.0;+SLC
rdm=10216331.837364519	145.2...0.254	dualstack	2014-02-06	16:54:25	Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT+6.1;+Trident/4.0;+SLC

Viele Fragen...

Ein Beispiel für einen Client, der sehr sicher OK ist aber aus bisher nicht geklärten Gründen die Zählpixel teilweise mit der selben ID angefordert hat:

cs_uri_query	c_ip	cs_host	date	time	cs_User_Agent	sc_status	
rdm=18934358.16074372	84.1	197	dualstack	2014-02-10	14:35:39	Mozilla/5.0+(Windows+NT+6.1;+WOW64;+Trident/7.0;+MATM;+rv:11.0)+like+Gecko	200
rdm=18934358.16074372	84.1	197	ipv4only	2014-02-10	14:35:39	Mozilla/5.0+(Windows+NT+6.1;+WOW64;+Trident/7.0;+MATM;+rv:11.0)+like+Gecko	200
rdm=16982878.30137952	84.1	197	ipv4only	2014-02-10	14:35:20	Mozilla/5.0+(Windows+NT+6.1;+WOW64;+Trident/7.0;+MATM;+rv:11.0)+like+Gecko	200
rdm=16982878.30137952	84.1	197	dualstack	2014-02-10	14:35:19	Mozilla/5.0+(Windows+NT+6.1;+WOW64;+Trident/7.0;+MATM;+rv:11.0)+like+Gecko	200
rdm=13839951.493950823	84.1	197	dualstack	2014-02-10	14:32:32	Mozilla/5.0+(Windows+NT+6.1;+WOW64;+Trident/7.0;+MATM;+rv:11.0)+like+Gecko	200
rdm=13839951.493950823	84.1	197	ipv4only	2014-02-10	14:32:32	Mozilla/5.0+(Windows+NT+6.1;+WOW64;+Trident/7.0;+MATM;+rv:11.0)+like+Gecko	200
rdm=17061247.221820797	84.1	197	dualstack	2014-02-10	14:32:24	Mozilla/5.0+(Windows+NT+6.1;+WOW64;+Trident/7.0;+MATM;+rv:11.0)+like+Gecko	200
rdm=17061247.221820797	84.1	197	ipv4only	2014-02-10	14:32:24	Mozilla/5.0+(Windows+NT+6.1;+WOW64;+Trident/7.0;+MATM;+rv:11.0)+like+Gecko	200
rdm=18389465.1450108	84.1	197	dualstack	2014-02-10	14:29:49	Mozilla/5.0+(Windows+NT+6.1;+WOW64;+Trident/7.0;+MATM;+rv:11.0)+like+Gecko	304
rdm=18389465.1450108	84.1	197	ipv4only	2014-02-10	14:29:49	Mozilla/5.0+(Windows+NT+6.1;+WOW64;+Trident/7.0;+MATM;+rv:11.0)+like+Gecko	304
rdm=18389465.1450108	84.1	197	dualstack	2014-02-10	14:29:40	Mozilla/5.0+(Windows+NT+6.1;+WOW64;+Trident/7.0;+MATM;+rv:11.0)+like+Gecko	200
rdm=18389465.1450108	84.1	197	ipv4only	2014-02-10	14:29:40	Mozilla/5.0+(Windows+NT+6.1;+WOW64;+Trident/7.0;+MATM;+rv:11.0)+like+Gecko	200
rdm=10727512.475233661	84.1	197	dualstack	2014-02-10	14:29:38	Mozilla/5.0+(Windows+NT+6.1;+WOW64;+Trident/7.0;+MATM;+rv:11.0)+like+Gecko	304
rdm=10727512.475233661	84.1	197	ipv4only	2014-02-10	14:29:38	Mozilla/5.0+(Windows+NT+6.1;+WOW64;+Trident/7.0;+MATM;+rv:11.0)+like+Gecko	304
rdm=10727512.475233661	84.1	197	dualstack	2014-02-10	14:29:35	Mozilla/5.0+(Windows+NT+6.1;+WOW64;+Trident/7.0;+MATM;+rv:11.0)+like+Gecko	304
rdm=10727512.475233661	84.1	197	ipv4only	2014-02-10	14:29:35	Mozilla/5.0+(Windows+NT+6.1;+WOW64;+Trident/7.0;+MATM;+rv:11.0)+like+Gecko	304
rdm=10727512.475233661	84.1	197	dualstack	2014-02-10	14:29:19	Mozilla/5.0+(Windows+NT+6.1;+WOW64;+Trident/7.0;+MATM;+rv:11.0)+like+Gecko	200
rdm=10727512.475233661	84.1	197	ipv4only	2014-02-10	14:29:19	Mozilla/5.0+(Windows+NT+6.1;+WOW64;+Trident/7.0;+MATM;+rv:11.0)+like+Gecko	200

Manchmal stimmt leider auch ein „not OK“



Dies ist ein Beispiel für einen Client, der scheinbar wirklich ein Problem mit dem AAAA-Record im DNS hat: die „guten“ Zugriffe sind zeitlich weit entfernt und auch ein anderer UserAgent ☹

cs_uri_query	c_ip	cs_host	date	time	cs_User_Agent	
rdm=10359783.72907266	194	5.110	dualstack	2014-01-12	20:51:26	Mozilla/5.0+(Linux;+U;+Android+4.1.2;+de-ch;+GT-I9100+Build/JZO54K)-
rdm=10359783.72907266	194	5.110	ipv4only	2014-01-12	20:51:26	Mozilla/5.0+(Linux;+U;+Android+4.1.2;+de-ch;+GT-I9100+Build/JZO54K)-
rdm=13564356.43043369	194	5.110	dualstack	2014-01-12	21:42:24	Mozilla/5.0+(Linux;+U;+Android+2.2.1;+de-de;+GT-S5830+Build/FROYO)-
rdm=13564356.43043369	194	5.110	ipv4only	2014-01-12	21:42:24	Mozilla/5.0+(Linux;+U;+Android+2.2.1;+de-de;+GT-S5830+Build/FROYO)-
rdm=10075208.756607026	194	5.110	ipv4only	2014-01-29	10:58:49	Mozilla/5.0+(Linux;+U;+Android+4.1.2;+it-it;+ME371MG+Build/JZO54K)+A
rdm=13498409.319484025	194	5.110	dualstack	2014-02-02	21:13:11	Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT+6.1;+Trident/4.0;+BT
rdm=13498409.319484025	194	5.110	ipv4only	2014-02-02	21:13:11	Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT+6.1;+Trident/4.0;+BT
rdm=13780299.117624027	194	5.110	dualstack	2014-02-08	12:13:53	Mozilla/5.0+(compatible;+MSIE+10.0;+Windows+Phone+8.0;+Trident/6.0;
rdm=13780299.117624027	194	5.110	ipv4only	2014-02-08	12:13:53	Mozilla/5.0+(compatible;+MSIE+10.0;+Windows+Phone+8.0;+Trident/6.0;
rdm=11918854.371178895	194	5.110	dualstack	2014-03-03	11:18:00	Mozilla/5.0+(Windows+NT+6.1;+WOW64)+AppleWebKit/537.36+(KHTML,
rdm=11918854.371178895	194	5.110	ipv4only	2014-03-03	11:18:00	Mozilla/5.0+(Windows+NT+6.1;+WOW64)+AppleWebKit/537.36+(KHTML,

Manchmal stimmt leider auch ein „not OK“



Dies ist ein Beispiel für einen Client, der scheinbar nicht in der Lage ist, mit einem DualStack DNS-Eintrag umzugehen ☹️

cs_uri_query	c_ip	cs_host	date	time	cs_User_Agent_
rdm=10106407.11571401	95. [redacted]	225 ipv4only	2014-01-20	07:45:53	Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT+5.1;+Trident/4.0;+GTB7.5;+.NET+CLR
rdm=19189270.569509983	95. [redacted]	225 ipv4only	2014-01-20	07:52:16	Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT+5.1;+Trident/4.0;+GTB7.5;+.NET+CLR
rdm=17424031.70144693	95. [redacted]	225 ipv4only	2014-01-20	07:53:10	Mozilla/4.0+(compatible;+MSIE+8.0;+Windows+NT+5.1;+Trident/4.0;+GTB7.5;+.NET+CLR

Manchmal will ein Client es mehrfach wissen

Dies ist ein Beispiel für einen Client, der die Zählpixel je dreimal anfordert 😊

cs_uri_query ↕	c_ip ↕	cs_host ↕	date ↕	time ↕	cs_User_Agent_ ↕
rdm=19248350.564020075	91 [REDACTED] 5.1	dualstack	2014-01-13	08:50:33	Mozilla/5.0+(compatible;+MSIE+9.0;+Windows+NT+6.1;+Trident/5.0)
rdm=19248350.564020075	91 [REDACTED] 5.1	dualstack	2014-01-13	08:50:33	Mozilla/5.0+(compatible;+MSIE+9.0;+Windows+NT+6.1;+Trident/5.0)
rdm=19248350.564020075	91 [REDACTED] 5.1	dualstack	2014-01-13	08:50:33	Mozilla/5.0+(compatible;+MSIE+9.0;+Windows+NT+6.1;+Trident/5.0)
rdm=19248350.564020075	91 [REDACTED] 5.1	ipv4only	2014-01-13	08:50:33	Mozilla/5.0+(compatible;+MSIE+9.0;+Windows+NT+6.1;+Trident/5.0)
rdm=19248350.564020075	91 [REDACTED] 5.1	ipv4only	2014-01-13	08:50:33	Mozilla/5.0+(compatible;+MSIE+9.0;+Windows+NT+6.1;+Trident/5.0)
rdm=19248350.564020075	91 [REDACTED] 5.1	ipv4only	2014-01-13	08:50:33	Mozilla/5.0+(compatible;+MSIE+9.0;+Windows+NT+6.1;+Trident/5.0)

- Bei **0.016%** eine Request-ID in mehr als 3 Log-Zeilen zu finden
es wurden 62 zufällige Request-IDs manuell überprüft und bewertet:
 - ❖ 74% der Clients sind OK (können mit A und AAAA DNS-Eintrag umgehen)
 - ❖ 26% der Clients sind Dual-Stack (IPv4 und IPv6 Adresse im Log)
 - ❖ Bei keinem einzigen der überprüften Requests ist ein Dual-Stack Problem zu vermuten
- (0.352%) **0.55%** eine Request-ID in genau 1 Log-Zeile (not-OK)
davon wurden 100 zufällige Zeilen manuell überprüft und bewertet:
 - ❖ 75% der Requests konnten unter anderem über IP-Adresse mit einem anderen Request verknüpft werden und der Client als OK definiert werden
 - ❖ 3% der Requests konnten auf diesem Weg mit einem IPv6 fähigen Client verknüpft werden
 - ❖ 22% der manuell überprüften Requests deuten darauf hin dass der Client wirklich vom Brokenness Problem betroffen ist

22% von 0.55% =

0.12% der Systeme unserer Kunden sind sehr wahrscheinlich vom Client Brokenness Problem betroffen

Basierend auf Zahlen vom März wären es 22% von 0.352% = 0.077%

Was würden wir beim nächsten Mal anders machen?



- Während der Vorbereitung des Vortrages die Gesamtstatistik im Blick behalten 😊
- Veränderung der Reihenfolge der Tracking-Pixel-Aufrufe ändern und Effekte beobachten
- Unterschiedliche Namen für die Zählpixel auf den verschiedenen Servern
 - v6-tracking1x1.gif für ipv6only
 - ds-tracking1x1.gif für dualstack
 - v4-tracking1x1.gif für ipv4only
- Request-ID mit PHP oder ASPX vom Server generieren lassen statt mit JavaScript auf dem Client generieren lassen
- Versuchen einen Zeitstempel in die Request-ID zu kodieren, eventuell auch die Client-IPv4-Adresse mit kodieren

Diese Seiten sind Startpunkte für weitere Informationen:

- Bin eventuell auch ich betroffen?
 - [http:// ip.bieringer.de](http://ip.bieringer.de)
 - <http://www.test-ipv6.com>Bitte unterschiedliche Browser und Geräte (auch Mobiltelefon und Tablet) testen
- Sehr ähnliche Statistik bei Eric Vyncke (<https://www.vyncke.org/testv6/>) und Tore Anderson (IPv6 dual-stack client loss in Norway, <http://www.fud.no/ipv6/>)
- IPv6-Tag bei heise mit Hintergrund-Infos (Aug.2010): <http://heise.de/-1066054>
- Kurze Auswertung des IPv6-Tag bei heise (Sept.2010): <http://heise.de/-1081201>
- World IPv6 Day (2011) und World IPv6 Launch Day (2012)
 - Meldung bei heise (Mai 2011): <http://heise.de/-1250059>
 - http://de.wikipedia.org/wiki/World_IPv6_Day_und_World_IPv6_Launch_Day

- Wer beschäftigt sich noch mit dem Thema „Messung“ von Client-Brokenness?
 - Ähnliches Prinzip?
 - Ähnliche Ergebnisse?
- Kennt jemand Client-Brokenness aus eigener Erfahrung?
 - Ursache?
 - Lösung?

Feedback und Austausch: ulrich.hauser@ifolor.ch;
gerne auch erst in ein paar Monaten mit eigenen Erfahrungen...



Vielen Dank!