464XLAT Optimization for CDNs/Caches

draft-palet-v6ops-464xlat-opt-cdn-caches

Jordi Palet (jordi.palet@theipv6company.com)
Alejandro D’Egidio (adegidio@telecentro.net.ar)
Problem Statement

• In IPv6-only networks using NAT46 (464XLAT, MAP-T), IPv4-only devices flows to dual-stack CDNs/Caches/services are terminated as IPv4, which means extra translations and the subsequent unnecessary overload.

• In equivalent IPv4-only CGN use cases, the CDNs provide “private” addresses (typically 100.64.0.0/10) to avoid exactly the same issues.
The NAT46/CLAT (WAN side) is connected by IPv6-only to the operator network, which in turn, will have a reverse function, the NAT64 ([RFC6146](https://tools.ietf.org/html/rfc6146)), known as PLAT (Provider Translator) in the case of 464XLAT. This allows to translate the IPv6-only flow back to IPv4, in order to forward it to Internet.

The translation of the packet headers is done using the IP/ICMP translation algorithm defined in [RFC7915](https://tools.ietf.org/html/rfc7915) and algorithmically translating the IPv4 addresses to IPv6 addresses following [RFC6052](https://tools.ietf.org/html/rfc6052).

In the case of 464XLAT, a DNS64 ([RFC6147](https://tools.ietf.org/html/rfc6147)) optionally is in charge of the synthesis of AAAA records from the A records, so they can use a NAT64, without the need of doing a double-translation by means of the CLAT. However, the DNS64 is not useful for the IPv4-only devices or applications in the LANs, as they will not be able to use the AAAA records.

A typical 464XLAT deployment is depicted in Figure 1.

![Figure 1: Typical 464XLAT Deployment](image-url)
IPv6-Capable device

Figure 3: 464XLAT access to CDNs/Caches by IPv6-capable apps
IPv4-only device

Figure 4: 464XLAT access to CDNs/Caches by IPv4-only apps
IPv4-only device (optimized)

Figure 5: Optimized 464XLAT access to CDNs/Caches by IPv4-only apps
In Summary

Optimal!

Dual-Stack User → ISP IPv6-only → Dual-Stack CDN/Cache
In Summary

Optimal!

Dual-Stack User → ISP IPv6-only → Dual-Stack CDN/Cache

Suboptimal!

Dual-Stack User → ISP IPv6-only → Dual-Stack CDN/Cache

CLAT 4→6

PLAT 6→4

ISP IPv6-only
In Summary

**Optimal!**

- Dual-Stack User
- ISP IPv6-only
- Dual-Stack CDN/Cache

**Suboptimal!**

- Dual-Stack User
- ISP IPv6-only
- Dual-Stack CDN/Cache

**Optimal!**

- Dual-Stack User
- ISP IPv6-only
- Dual-Stack CDN/Cache

---

ISP IPv6-only only for Dual-Stack CDN/Cache

PLAT 6->4

CLAT 4->6

Optimal!

Suboptimal!

Optimal!
Approach 1: DNS/Routing-based

- CLAT translate A records into AAAA:
  - WKP::A or NSP::A
- CDN/Cache provider configures dedicated interfaces to match WKP::A or NSP::A

<table>
<thead>
<tr>
<th><a href="http://www.example.com">www.example.com</a></th>
<th>A</th>
<th>192.0.2.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLAT translated to</td>
<td></td>
<td>64:ff9b::192.0.2.1</td>
</tr>
<tr>
<td>CDN IPv6 interface must be</td>
<td></td>
<td>64:ff9b::192.0.2.1</td>
</tr>
<tr>
<td>Operator must have a specific route to</td>
<td></td>
<td>64:ff9b::192.0.2.1</td>
</tr>
</tbody>
</table>

- Issues:
  - Only works if “local/private” connectivity
  - CDN/Cache provider needs to do “something”
Approach 2: CLAT/DNS-proxy-EAMT

- NAT64/CLAT/CE is also a DNS proxy/stub resolver, so an internal interaction can be created.
- This approach uses existing IPv4 and IPv6 addresses (A, AAAA RRs), so no additional complexity for services.

Steps:
- Detection of IPv4-only devices or apps
- Detection of IPv6-enabled service
- Creation of EAMT entries
- Forwarding path for existing EAMT entries
- Maintenance of the EAMT entries
Approach 2 Example

- Example

www.example.com  A  192.0.2.1
    AAAA  2001:db8::a:b:c:d

EAMT entry  192.0.2.1  2001:db8::a:b:c:d

NAT64/CLAT translated to  2001:db8::a:b:c:d

CDN IPv6 interface already is  2001:db8::a:b:c:d

Operator already has specific route to  2001:db8::a:b:c:d

1. A query for www.another-example.com A RR is received
2. www.another-example.com  A  192.0.2.1
4. A conflict has been detected
5. The existing EAMT entry for 192.0.2.1 is set as invalid
Approach 2: Additional Considerations

- Behavior in case of multiple A/AAAA RRs
- Behavior in case of presence/absence of DNS64
- Behavior when using literal addresses or non IPv6-APIs
- False detection of a dual-stack host as IPv4-only
- Behavior in presence of HE
- Behavior in case of Foreign DNS
  - Devices/apps using other DNS
  - DNS privacy/encryption
  - DNS modified by user in OS
  - DNS modified by user in CE
  - Combinations of above
Approach 3: CLAT-provider-EAMT

- Similar to previous one, but no "automated" EAMT
- Operator must push or CE must pull the table
- It will work even if user change DNS for STB, SmartTV, …
- More control from the operator
  - EAMT pairs may be built “apart” from DNS
- Issues:
  - Increase complexity
    - Is the benefit worth for it?
  - Need to add TTL (from DNS) to EAMT
Solution for IPv6-only Services?

- **NOT POSSIBLE!** ---&gt; NO IPv4 connectivity

---

**Optimal!**

Add A RRs even if IPv4 is not available
Next steps

• Become a WG item?

• Title change?

• New inputs? CDN/cache providers, others