464XLAT/NAT64 Optimization

draft-ietf-v6ops-464xlat-optimization-02

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 many cases this may become a show-stopper
- In equivalent IPv4-only CGN use cases, the CDNs accept “private” addresses (typically 100.64.0.0/10) to avoid exactly the same issues

Typical 464XLAT Deployment

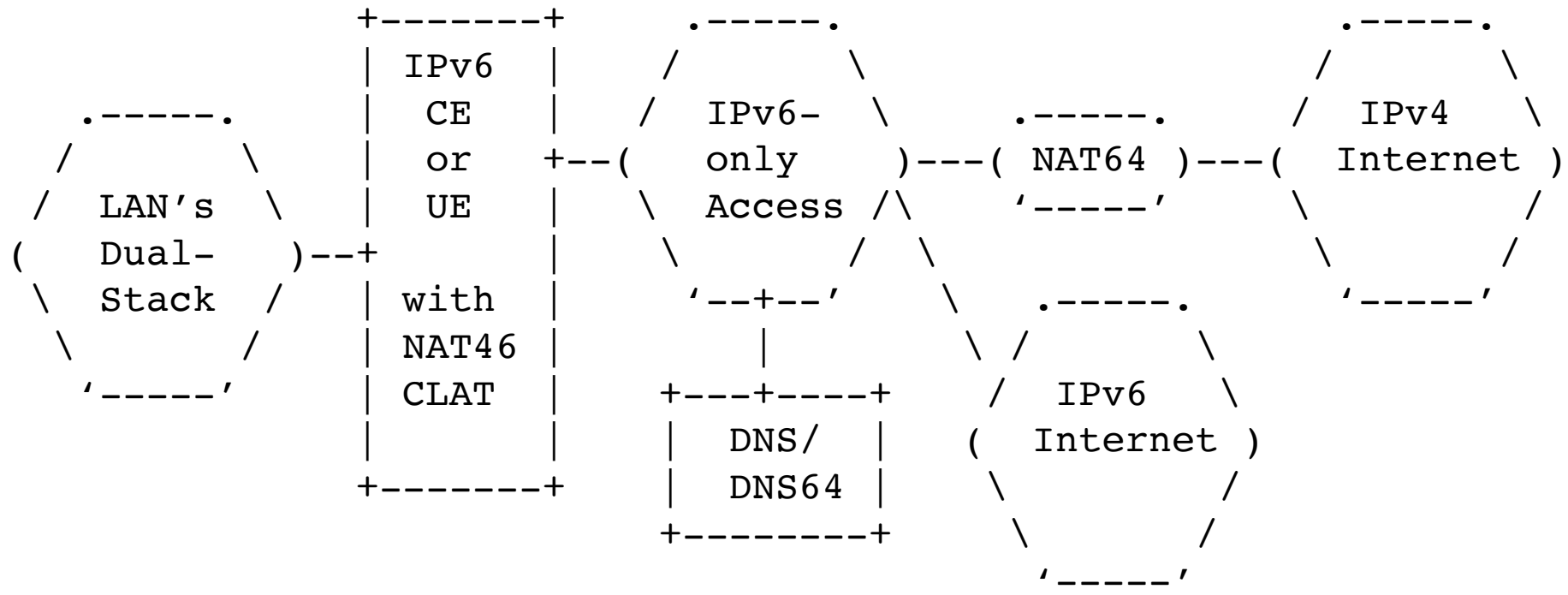
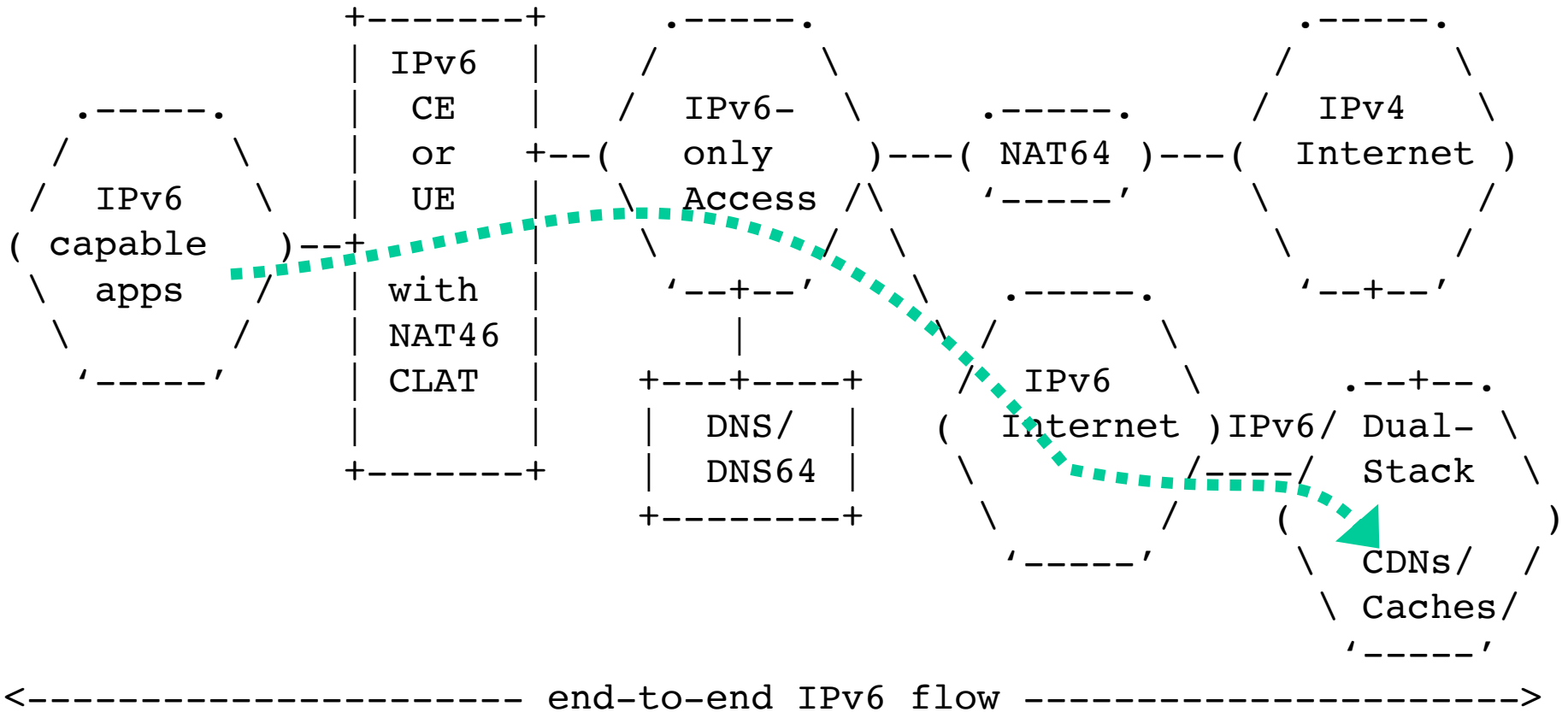


Figure 1: Typical 464XLAT Deployment

IPv6-Capable device



IPv4-only device

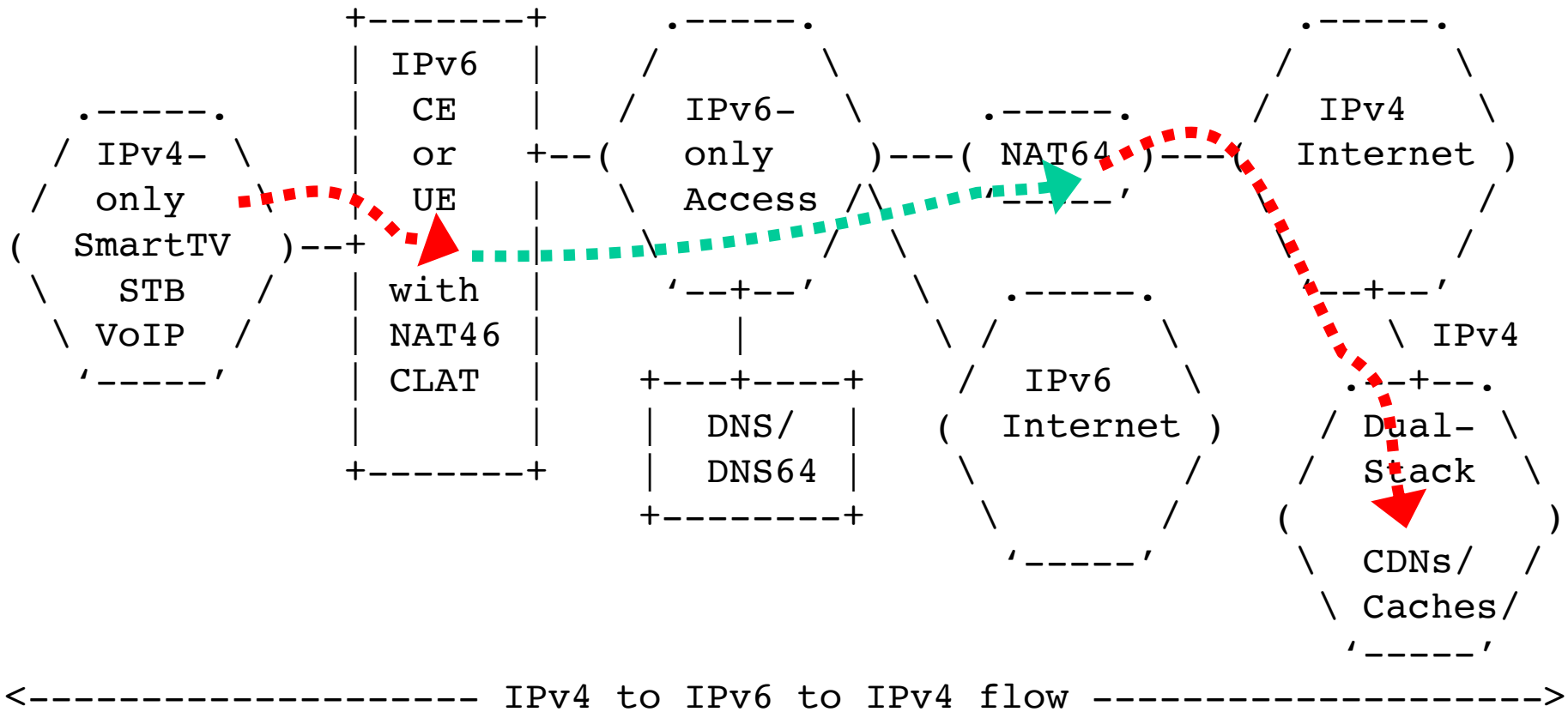
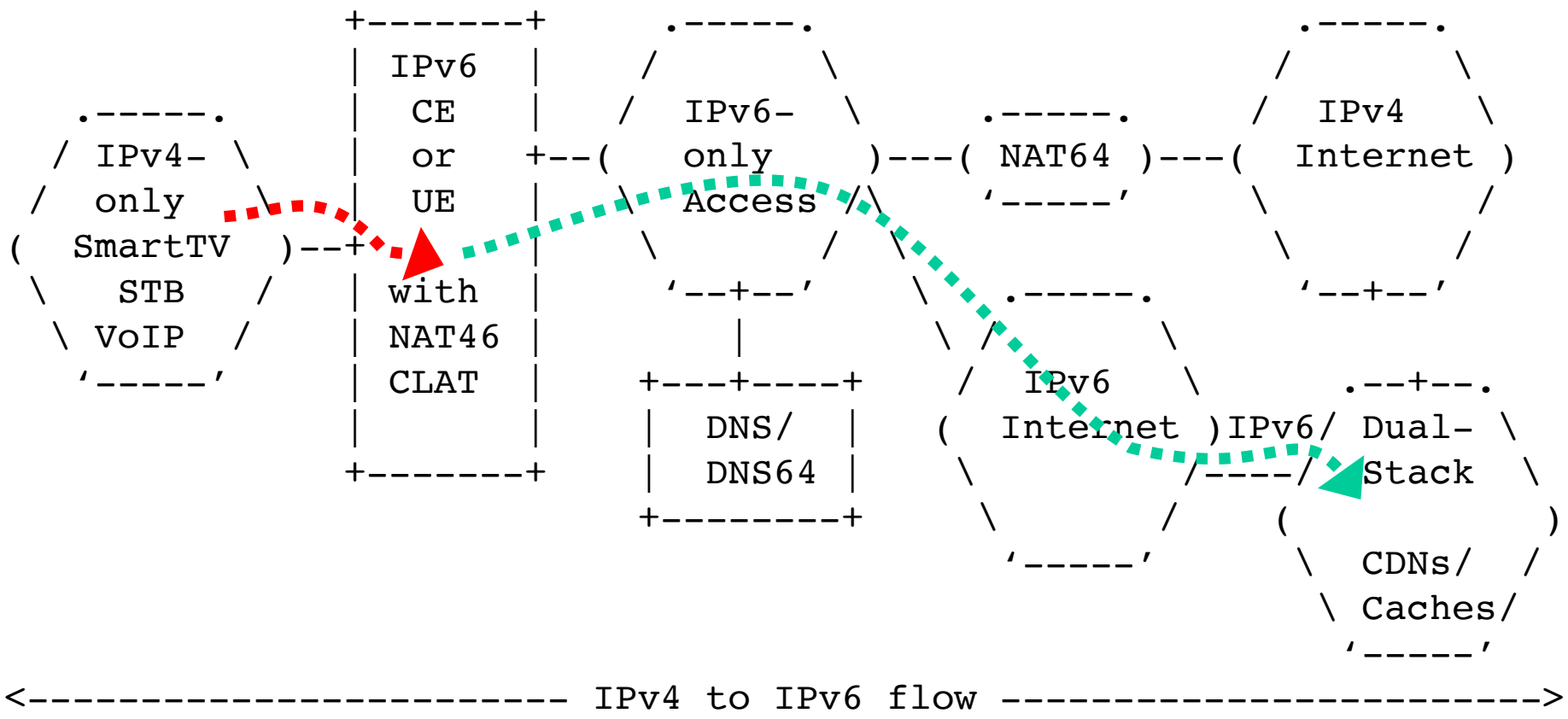


Figure 4: 464XLAT access to CDNs/Caches by IPv4-only apps

IPv4-only device (optimized)



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

www.example.com	A	192.0.2.1
CLAT translated to		64:ff9b::192.0.2.1
CDN IPv6 interface must be		64:ff9b::192.0.2.1
Operator must have a specific route to		64:ff9b::192.0.2.1

- Issues:
  - Only works if “local/private” connectivity
  - CDN/Cache provider needs to do “something”

Approach 2: CLAT/DNS-proxy-EAMT

- NAT46/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
3. www.another-example.com AAAA 2001:db8::e:e:f:f
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

Update from previous version

4.2.4. Forwarding path via stateful NAT for existing EAMT entries

Following this approach, if there is a valid EAMT entry, for a given IPv4-destination, the IPv6-native path pointed by the IPv6 address of that EAMT entry, will take precedence versus the NAT64 path, so the traffic will not be forwarded to the NAT64.

However, this is not sufficient to ensure that individual applications are able to keep existing connections. In many cases, audio and video streaming may use a single TCP connection lasting from minutes to hours. Instead, the CDN TTLs may be configured in the range from 10 to 300 seconds in order to allow new resolutions to switch quickly and to handle large recursive resolvers (with hundreds of thousands of clients behind them).

Consequently, the EAMT entries should not be used directly to establish a forwarding path, but instead, to create a stateful NAT entry for the 4-tuple for the duration of the session/connection.

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

Pending discussion from the list

- Jen Linkova assigned as shepherd, 1st round of edits already applied into -02
- Other inputs received from:
  - Jen Linkova
  - Erik Nygren
  - Vasilenko Eduard
  - Erik Kline
- Pending discussion/responses about:
  - Ensuring that no breakage is generated
  - Security (DNS, others)
  - Clients caching, TTLs
  - …

Additional topic: RFC6877 to standard

- It is clearly the transition mechanism that has *more* users and going up
- However, is not a “standard”
- Should we work into progressing it?
- Is something that could be done in v6ops?