# Testing Applications' IPv6 Support

`draft-tiesel-v6ops-ipv6-app-testing-01`


| | |
|---|---|
| Philipp Tiesel | philipp.tiesel@sap.com |
| Jen Linkova | furry@google.com |
| Kyle Ouellette | kouellette@iol.unh.edu |
| Ben Patton | bpatton@iol.unh.edu |

# The advent of IPv6-only for applications

US Governmental Mandate M21-07

*Completing the Transition to Internet Protocol Version 6*

creates pressure in large parts of the industry to support IPv6

... it also re-defines what “supporting IPv6” means.

# What “supporting IPv6” means.

*The application SHOULD NOT break in presence of IPv6 and MAY prefer IPv6 for its communication*

*The application MUST work in an IPv6-only environment without functional regressions*

*... do you have something on testing application for IPv6?*

# *There was nothing*

So...

- Decided that this needs to become a BCP
- `draft-tiesel-v6ops-ipv6-app-testing-00` released while on IETF 123 in Madrid
- Found co-authors with a different perspective
- Produced a joint -01 and present it here

# Objectives of IPv6 app testing draft

- Provide concise definition what “supporting IPv6” means.
- Define “Base Connectivity Scenarios” as basis for testing
- Give guidance on how to carry out IPv6 testing for applications including lifecycle aspects of the application
- Discuss Common Sources of IPv6 Related Failures and Misbehavior

*... now we come to the difficult parts*
*... and things we avoided*

# Definition of “Supporting IPv6” (1/2)

*Defining “Supporting IPv6 is not only difficult,*
*it also makes IPv6 that strange Unicorn thing*
*developers are supposed to herd from now on*
*in addition to their other tasks…*

# Definition of “Supporting IPv6” (2/2)

Instead, we define a target state:

> *Applications are expected to function*
> *regardless of whether they are used in*
> *an IPv4-only environment,*
> *a Dual-Stack environment, or*
> *an IPv6-only environment,*
> *with or without connectivity to the IPv4 Internet.*

# Base Connectivity Scenarios (1/2)

*Defining them brought us into the unfavorable situation*

*to clearly define what “IPv6-only” means...*

# Base Connectivity Scenarios (2/2)

Instead, we defined four other things:

- IPv4-only
- Dual-Stack
- IPv6-only with NAT64
- True IPv6-only

# Core Testing Objective

IPv6 application testing should always be derived from functional and integration testing.

Therefore, the goal is to verify that the expected behavior is consistent across all connectivity scenarios, i.e., the application functions correctly in IPv4-only, Dual-Stack, IPv6-only with NAT64 and True IPv6-only settings.

# Connectivity Scenario Configurations

- Start with a client/server assumption
- Take the full combination matrix
- Explain which combinations need testing anyway
- Explain which scenarios may be skipped when and why

- Considerations for Intermediaries and broken connectivity

| Client | Server | Classification |
|---|---|---|
| Dual-Stack | IPv4-only | base |
| Dual-Stack | True IPv6-only | base |
| IPv4-only | Dual-Stack | base |
| IPv6-only with NAT64 | IPv4-only | base |
| True IPv6-only | Dual-Stack | base |
| IPv4-only | IPv6-only with NAT64 | IPv6-only-DC |
| Dual-Stack | Dual-Stack | extended |
| IPv4-only | IPv4-only | extended |
| IPv6-only with NAT64 | True IPv6-only | extended |
| True IPv6-only | True IPv6-only | extended |

*Table 1: Connectivity scenario combinations to consider*

# Testing Strategies

- Best practices how to approach IPv6 testing
- How to test if a full connectivity combination matrix testing is infeasible

→ Call to action: What did we miss?

# Common Sources of IPv6 Related Failures and Misbehavior

- Collection of things to watch out

→ Call to action: What did we miss?

# Next steps

- Let’s start the discussion now
- We would love to get feedback,
  especially from different perspectives

- Is this valuable and is the WG interested in the topic?
  → Call for adoption