PANAPI, Milestone 3

Thorben Krüger

Networks and Distributed Systems Lab
Otto-von-Guericke-Universität
Magdeburg

October 7, 2022

Outline

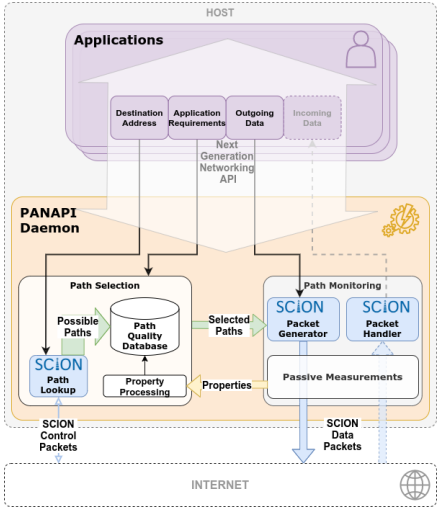
1. Project Overview
2. Project Status

PANAPI - Design

- Written in Go
- Backend scriptable in Lua
- Support for TCP/UDP/QUIC as well as SCION with (UDP/QUIC)
- Frontend API follows the IETF TAPS architecture as much as possible

PANAPI - Backend

- Currently SCION-specific
- Collects and caches network performance information per Path ✅
- Scriptable path quality estimation
  - scriptability ✅
  - scripts themselves ✅
- Scriptable path choice per Message/Packet ✅
- Scriptable active latency measurements per Path (stretch goal ☑)



Project Milestones - Overview

- Milestone 1: Ethics Approval & Kick Off ✅
- Milestone 2: Basic Path Selection ✅
- Milestone 3: Advanced Path Selection ✅
- Milestone 4: Evaluation of Path Selection ✅

Milestone 3: "Improving the MVP"

KPI Description

Implementation of advanced path selection features in scope of an example application

KPI Means of Verification

Demonstration of the example application on top of the full-fledged path selection implementation

Milestone 3: Results

Implementation of advanced path selection features

- Completed ✅
- Very Powerful Scripting API ✅
- `https://github.com/netsys-lab/panapi`

Example application making use of advanced path selection features

- Completed ✅
- Takes the form of a `Capacity Seeking` Bandwidth Benchmarking App

```
function nextBestBWPath(laddr, raddr)
   local fp = laddr2fp[laddr]
   local fingerprints = {}
   for _, fp in ipairs(raddr2fps[raddr]) do
      table.insert(fingerprints, fp)
   end
   table.sort(
      fingerprints,
      function(fp_a, fp_b)
         return (fp2bw[fp_a] or 1000) < (fp2bw[fp_b] or 0)
      end
   )
   return fp2path[fingerprints[1]]
end


-- gets called when a set of paths to addr is known
function panapi.Initialize(prefs, laddr, raddr, ps)
   panapi.Log("New connection [" .. laddr, "|", raddr .. "]")
   raddr2fps[raddr] = raddr2fps[raddr] or {}
   laddr2switchtime[laddr] = panapi.Now()
   for i, path in ipairs(ps) do
      local fp = path.Fingerprint
      fp2path[fp] = path
--       fp2last[fp] = tick
      fp2id[fp] = i
      panapi.Log("Path", i, fp)
      table.insert(raddr2fps[raddr], fp)
   end
   panapi.SetPreferences(prefs, laddr, raddr)
end
```

Demo