Performance Verification Architecture Task Force

Update

Jerry Sobieski (NORDUnet)
Steve Wolff (Internet2)
Jan 17, 2013
Honolulu, HI US

NORDUnet
Nordic infrastructure for Research & Education

Performance Verification Architecture

- Objective (from the charter):
  - The GLIF End-to-End Performance Verification Architectures task force is chartered to develop recommendations for a deterministic, scalable, and secure architecture for determining the delivered end to end performance characteristics of emerging light path (connection oriented) network services.
- Co-Chairs
  - Steve Wolff (Internet2)
  - Jerry Sobieski (NORDUnet)

Progress

- Discussion began in early 2012…
- Limited participation… (still not sure why…)
- Basic conference call: "Why can't we just use perfSONAR?..."
- …need a paper describing future services, why verification is important in this future environment, what needs to be verified, when, where, how by whom,… What capabilities will a comprehensive and deterministic PV process require?
- Paper is [almost] available:
- Purpose: Explain why verification of performance *guarantees* for Connection Services is different from characterizing conventional best effort performance – and why deterministic methods are necessary not just to insure delivery of predictable performance, but to understand how such services behave – or fail.

  → Get GLIF community to recognize why PV is critical for the success of guaranteed services.
- **"Performance Verification Architecture and Emerging Performance Guaranteed Network Services"**
  - …some editing is still in progress (18 pages + illustrations).
  - Will be circulated for review before Jan 2013.
- Summary follows…

Deterministic End to End Performance Verification Architecture

Offered for discussion to the GLIF Performance Verification Architectures Task Force

Jerry Sobieski
NORDUnet
Jan 17, 2013
Honolulu, HI US

NORDUnet
Nordic infrastructure for Research & Education

Performance

"Measure what can be measured, and make measurable what cannot be measured."

- Galileo Galilei

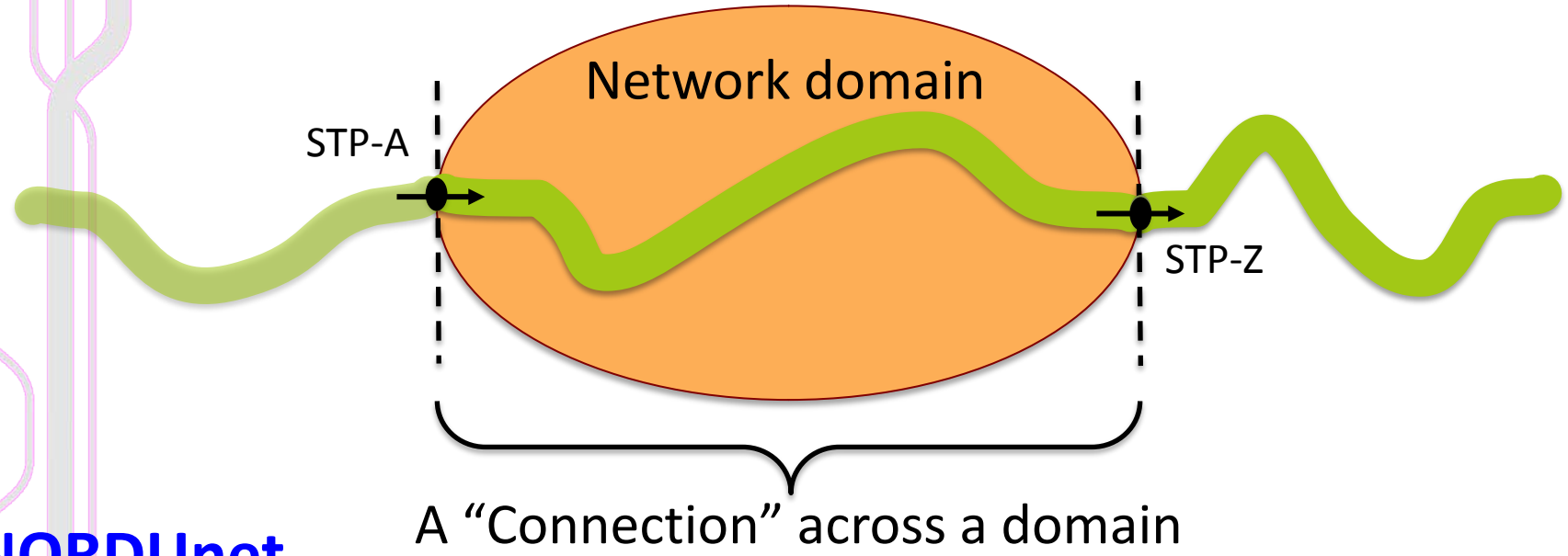
- Definition: Performance is a possibly multivariate quantity that can be measured to an agreed-upon precision by an agreed-upon measurement protocol.

The Problem

- Emerging Connection Services offer “guaranteed” performance …or so they say.
- How do we verify this performance?
  1. Determining when a Connection is performing as requested/required…or not.
  2. Determining which aspects of the performance guarantees are not functioning to spec
  3. Determine (to some resolution) “where” a Connection is failing
- Guaranteed services are intended to provide deterministic service – predictable, reliable, repeatable… And so require substantially tighter engineering constraints than best effort
  - Deterministic PV processes are critical
- If performance is flawed, it needs to be fixed, ASAP.
  - Identifying the under-performing segment and notifying the agent in charge via automated means is End Game (Automated Fault Localization)

What are we "verifying"?

- What networks do: transport user data from one point to another.
- Performance guarantees are network services that make that transport behavior *predictable* and *repeatable*.
- Connections are "logical conduits that carry user data transparently and unmodified from an ingress location to an egress location" (ref: OGF NSI Framework 2011)



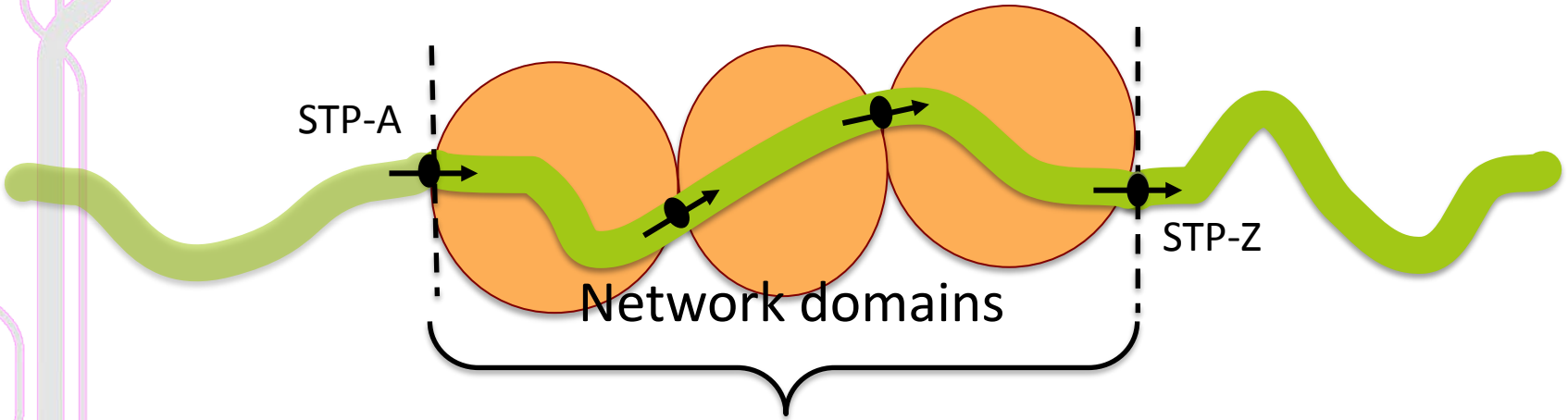
NORDUnet
Nordic infrastructure for Research & Education

Basic Performance Verification Tests

- Bits went in, … did they come out?
  - Did any bits NOT come out? (lost/discarded data)
  - Did bits come out that did not go in? (mis-routed data)
- Since user data must be serialized for transport (e.g. large files/data sets), or is time sensitive intrinsically (e.g. video), there is a natural temporal aspect to guaranteed services:
  - Capacity/good put: Bits *PER SECOND* (or “per unit time”)
  - Latency: time the bits spend in the transport medium
  - Jitter: the variance in latency (time) for datagrams transiting the transport conduit
- Thus, PV needs to characterize the behaviour of the transport payload with respect to time
  - What was the the [payload] transmission schedule? What was the observed arrival schedule… How do they relate to one another?
  - The better the time resolution of these schedules, the better the understanding of network performance will be.

Global *inter-Domain* Services

- In the real world…End to End connection services are predicated across a multiple network service domain…
- "Inter-operating common services domains" are not equivalent to "a single administrative domain"…
  - These are separate and independent organizations and services
  - With local realities: levels of expertise, performance capabilities, authorization constraints, security/privacy concerns, accounting, legal environments, peering arrangements,…



How do we verify the performance end to end?

Service Instance and Service Definition

- A Connection is an Instance of a Connection Service
- A Connection Service has specific set of service attributes it recognizes and can provision and guarantee.
  - Examples:
    - Capacity = 0 to 100 Gbps
    - Max Frame Error Rate = 1 *10^-8
- Some conventional service attributes are are difficult or impossible to measure – as traditionally stated:
  - Example: Capacity
    - "5 Gbits per second" connection over a 10 Gbps link layer…
    - is that "10 gbits for half second followed by quiesence for half second."?
    - Why not 10 gbits for 10 seconds followed by 10 seconds of quiet"?
- The Service Definition needs to be specific, explicit, …
- And *measureable*

- Why ask for a service instance with an attribute that cannot be "felt" by the user…cannot be detected or measured.

Why verify?

- As Users, we specify performance requirements in advance so that we can predict how our application will behave …
  - Perhaps we want to know that our file transfer will complete by a certain deadline,…
  - Or we want to be certain that our video streams are of the highest quality (unaffected by unrelated traffic)
  - Or perhaps we are building our own virtual network and we want a known capacity or latency for our own purposes…
- Performance guarantees [*are supposed to*] provide a predictable *deterministic* service.

Deterministic Verification

- *Deterministic* means that given a set of known initial conditions, the result of a process is pre-determined…it will not vary
  - It is predictable
  - It is *reliably* predictable
  - It is repeatable
- For performance guaranteed connection services, the performance verification process should also be deterministic
  - Given the same data flow, the same performance will be observed, every time.

Why verify?

- As Users, we do not trust the provider.
  - The provider is incompetent – despite their best intentions and efforts, they can't get it right…
  - The provider is a shyster – the PA knowingly does not enforce the performance guarantees ("we have 100 Gbps core – we don't need to enforce bandwidth constraints in the core...")
  - The service itself is ill-defined wrt certain performance aspects
  - s#!t happens – something outside of the control of the provider broke.
- The user needs an *independent* means of determining whether the service they received meets the specifications they requested and were promised.

Why verify?

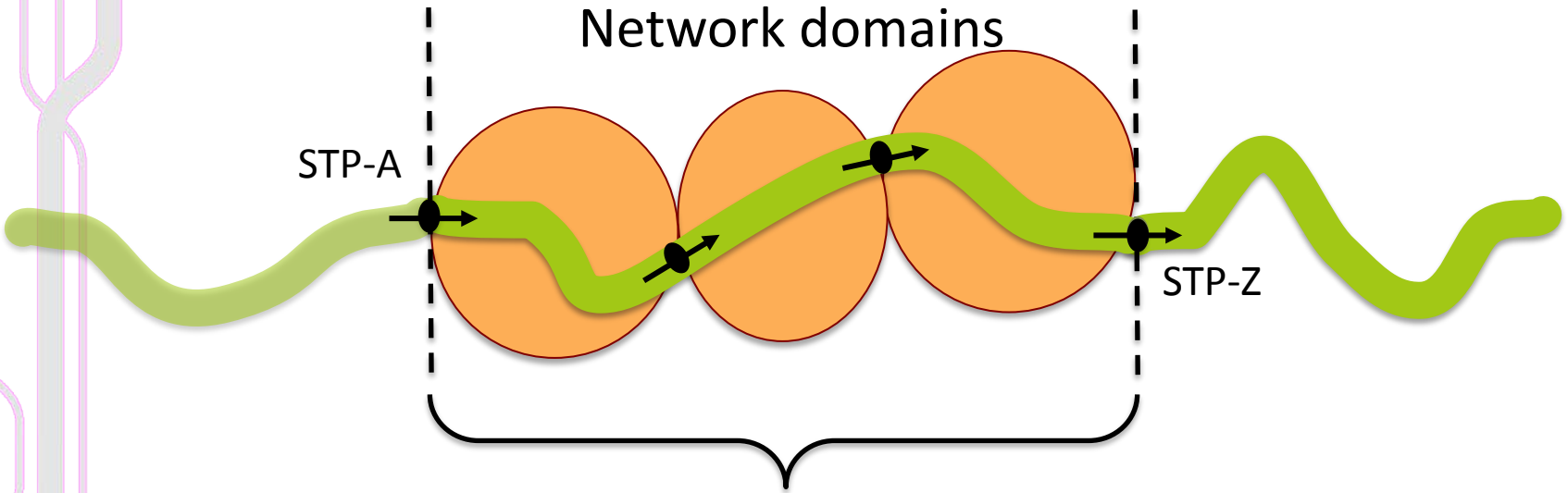
- As Providers, we do not trust the user:
  - Users are incompetent – they do not configure their end system(s) or access networks to properly interact with the provisioned performance capabilities.
  - The user has mis-interpreted the capabilities and is not conforming to the service guarantees (or the service is undefined with respect to some aspect.)
  - s%#$@t happens – something outside of our control broke – we can not rely on the user to notify us prior to the law suit.
- The Provider needs an independent means of determining whether the service they delivered meets the specifications they guaranteed

Independent Verification

- "Verification" means corroborating what the other party claims to be true.
- "Independent verification"
  - the user's verification testing does not rely on their provider assets/agents to provide corroboration
  - And the provider's tests likewise do not rely on the user's assets or agents for its results.
- The PV architecture must be able to provide an independent testing/measurement model.

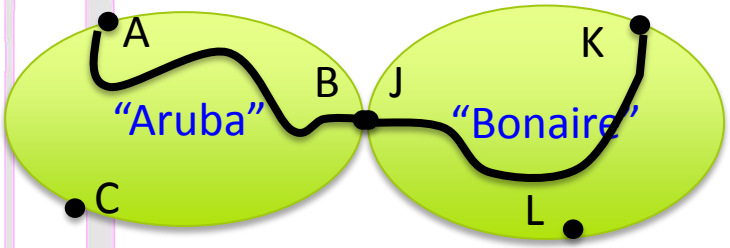
Deterministic Performance Verification

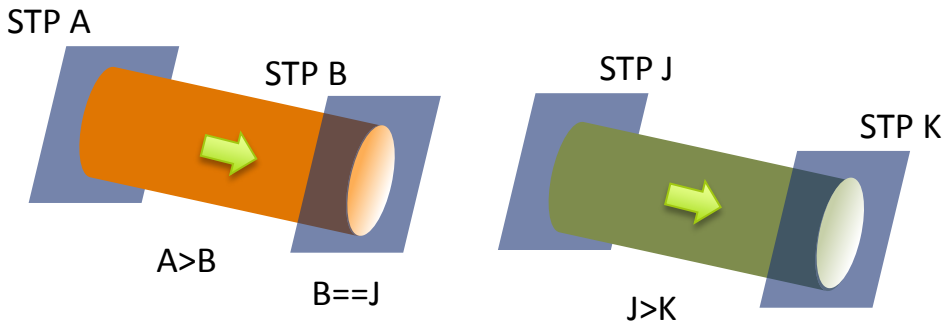
- Can we *deterministically* measure the performance of a Connection?
- Can we do so without perturbing the flow?
- Can we do so in such a fashion that we can determine where along the path performance problems are occuring?

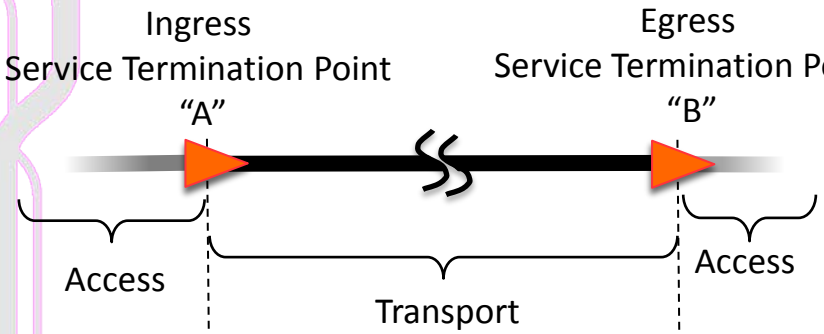


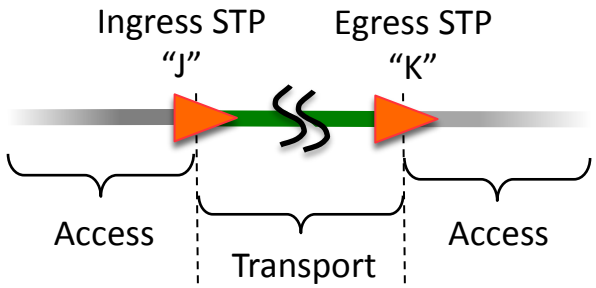
How do we verify the throughput from STP-A to STP-Z?

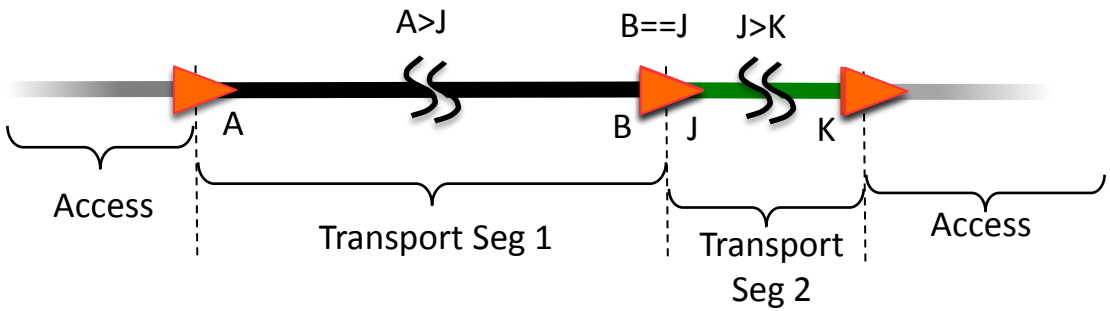
Connection Segmentation





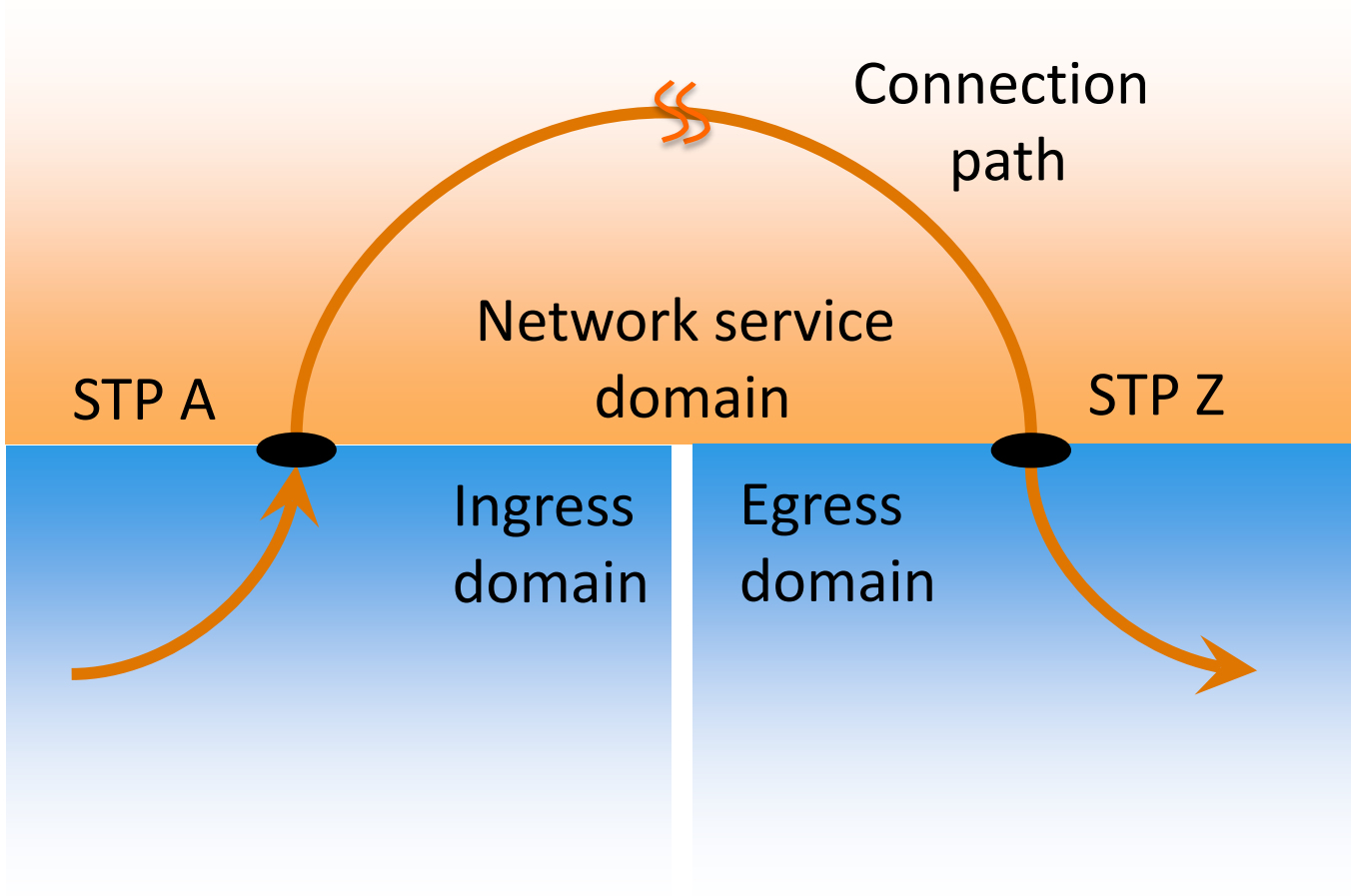




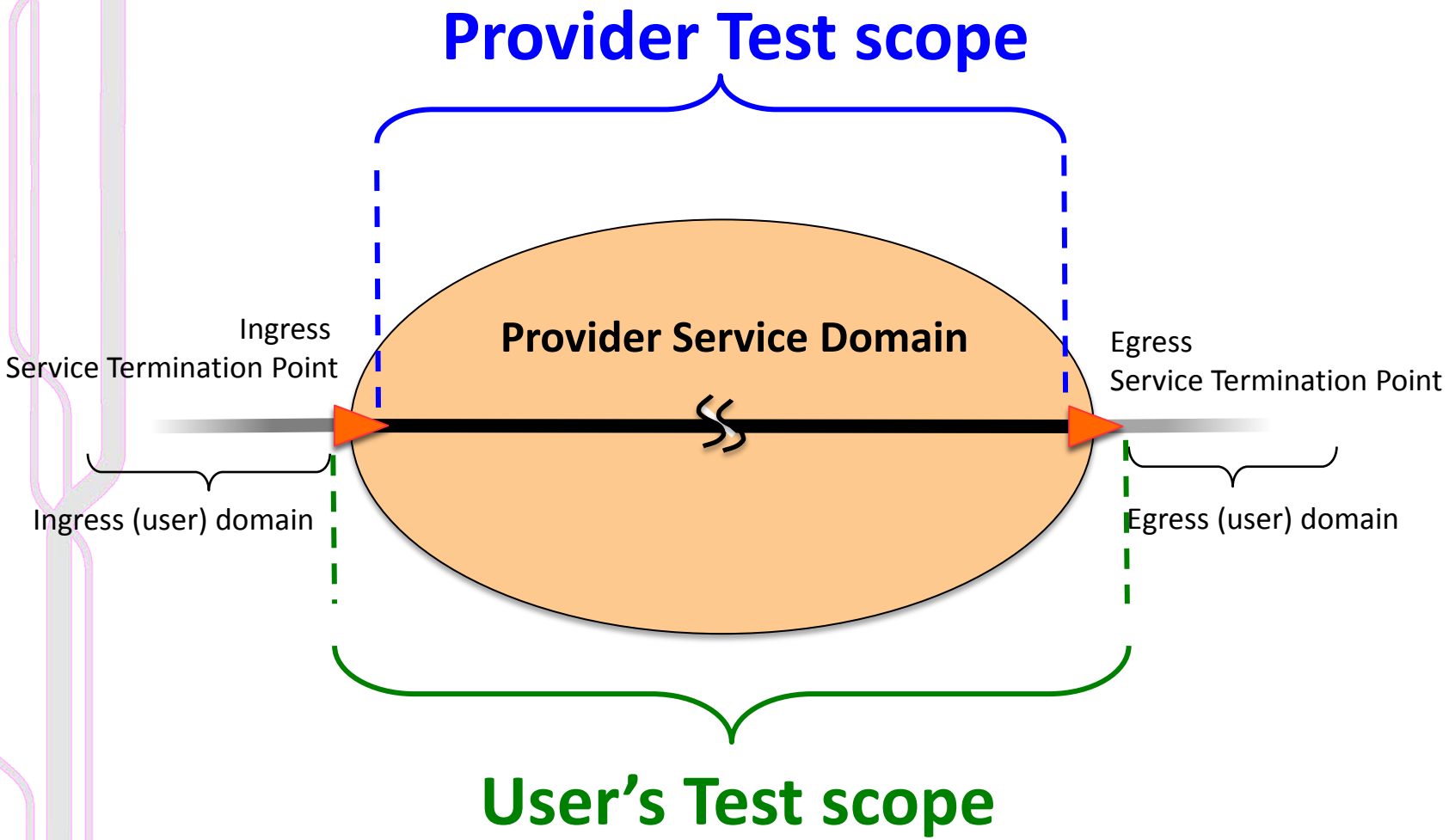


NORDUnet
Nordic infrastructure for Research & Education

The Service Provider Demarc



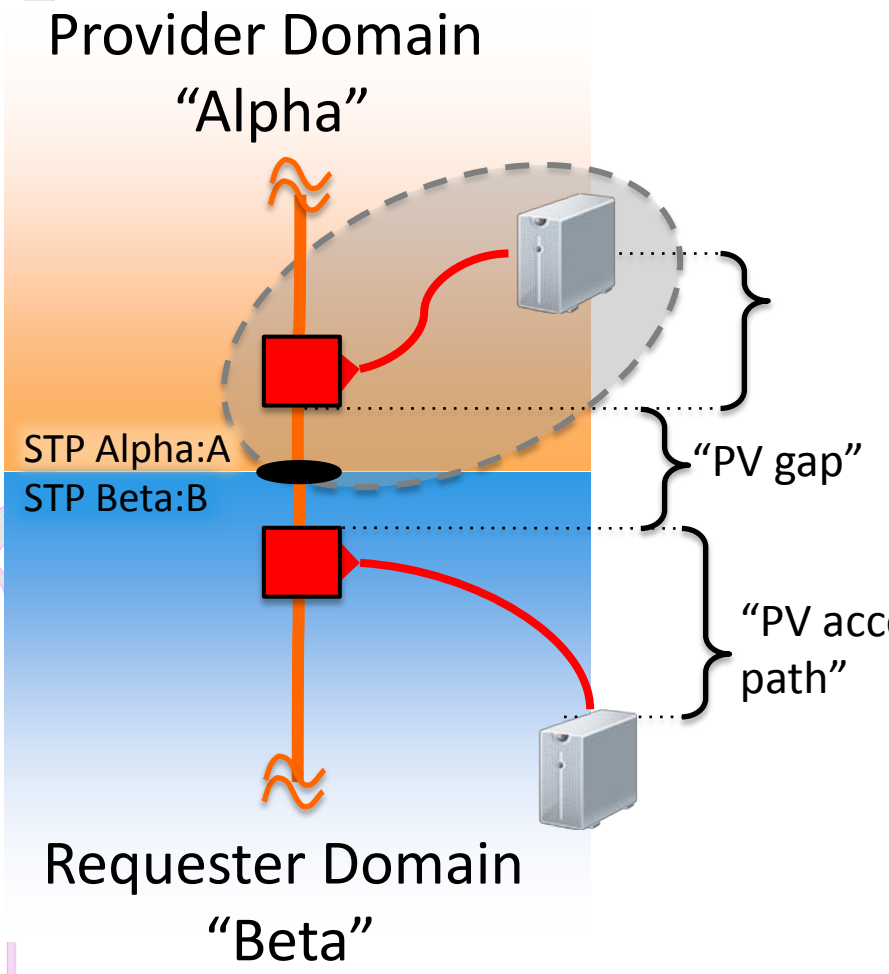
PV “Test Scope”



NORDUnet

Nordic infrastructure for Research & Education

PV Test Point Engineering

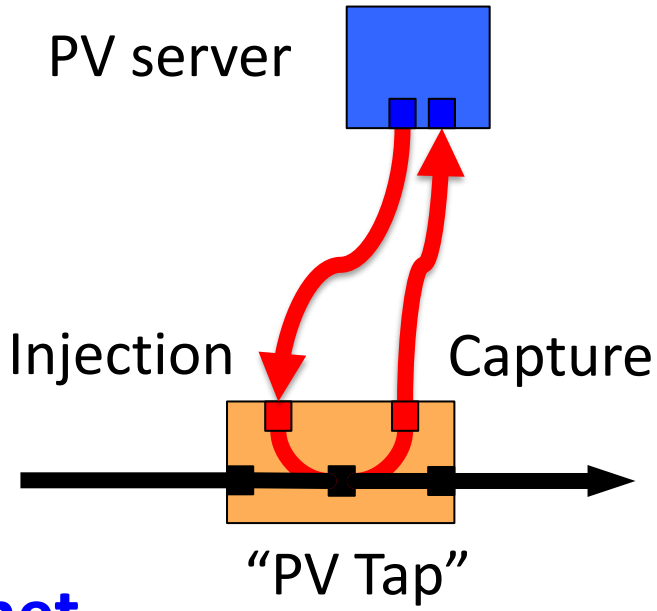


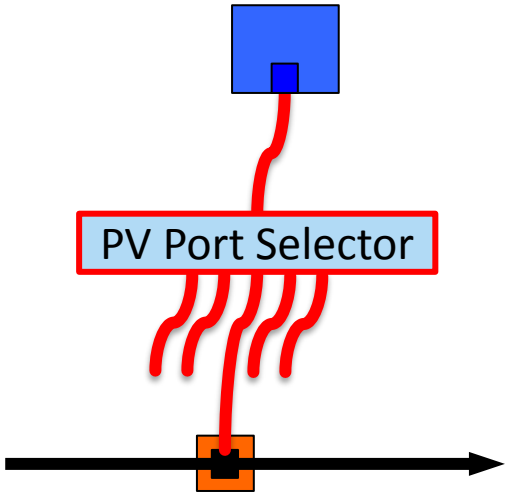
"PV Test Points" must be: situated as physically close to the demarcation STP as possible

- Minimize effects of difference ("gap") between PV test points in adjacent domains
- Compact PV access in order to minimize path effects introduced by PV infrastructure itself

PV Test Point Infrastructure

- "Tap" – A device in the data plane path that allows traffic to be inserted or observed.
- "PV Server" – an intelligent device that can source or sink data flows via the PV tap.
- PV Interface – the data channel between the PV Tap and the PV Server. (Note this may not be a simple patch cable)





NORDUnet
Nordic infrastructure for Research & Education

Notes on PV Test Point Engineering

- The Test Point “tap” is very inexpensive (scalable)
  - An optical splitter, or low cost UTP
- Many [high end] switches and routers already have similar functionality built into interfaces as “loopback” features…
- In some cases, the hardware technology already supports a similar “tap” capability:
  - E.g. Ethernet “span” ports…
  - The implementation of these features may induce or inject other artifacts into the PV path and should be well understood if used for this purpose
  - …but can provide almost free proof of concept

(Note: this “tap” technology has been used in various firewalls and monitoring products for many years.)

Notes on PV Test Point Engineering

- The PV Server contains more intelligence
  - The PV hardware interface NIC must be able to *at least* capture flows…thus must recognize 802.1, SONET/SDH, infiniband, or any other physical layer transport protocol
- The device must be able to capture the flow to long term storage at line rate
  - 100 Gbps (!?)
  - Performance analysis is done as a background process non-real time
- For emergent services – the capture process should anticipate multi-protocol flow identifiers

Notes on PV Test Point Engineering

- Timing: The interface must be able to time-stamp the flow appropriately, and accurately.
  - For asynchronous 10 Gbps packet networks (ethernet?), 50 byte (500 bits@8b/10b encoding) resolution would require ~50 nanosecond timestamp resolution
  - Jitter measurements would dictate somewhat smaller resolution order 1-5 nanoseconds
  - For 100 Gbps links, divide by 10… ~ 100 picosecond stamping.
- Accurate timestamps at this resolution will likely require external clocks
- End-to-end Latency timing will also need new technology (satellite based sync will not likely be sufficient for future services)

Notes on PV Test Point Engineering

- Scaling
  - The test device can be shared – e.g. one PV server for a bank of physical ports…
  - Switchable optronics can mate the port under test with the PV server with minimal path artifacts.
- Common PV Test Elements
  - Two peering networks could agree to share a test point
  - The trust issue can be resolved by verifying the performance of the test point itself
    - Once verified, the PV Test point can be “trusted”
    - “Mutually trusted third party” test points
  - This poses some challenges to the segmented test process…but it can still be deterministic

Performance Flow Correlation

- Understanding the behaviour of jitter, latency, and good put of a guaranteed service (“connection”) requires comparing the observed egress flow to the known ingress flow characteristics – “*flow correlation*”
- Deterministic PV captures and characterizes the source flow as tightly as the destination flow, and then the two flows are compared.
  - The comparison requires one or both captured flows be transferred to another host for correlation.
- Near real-time results can be had for most tests (delays of only seconds or minutes for large flow analysis.)
  - Flow correlation processing (transferring ingress and egress flows) must not interfering with test flows themselves.

Passive Performance Verification

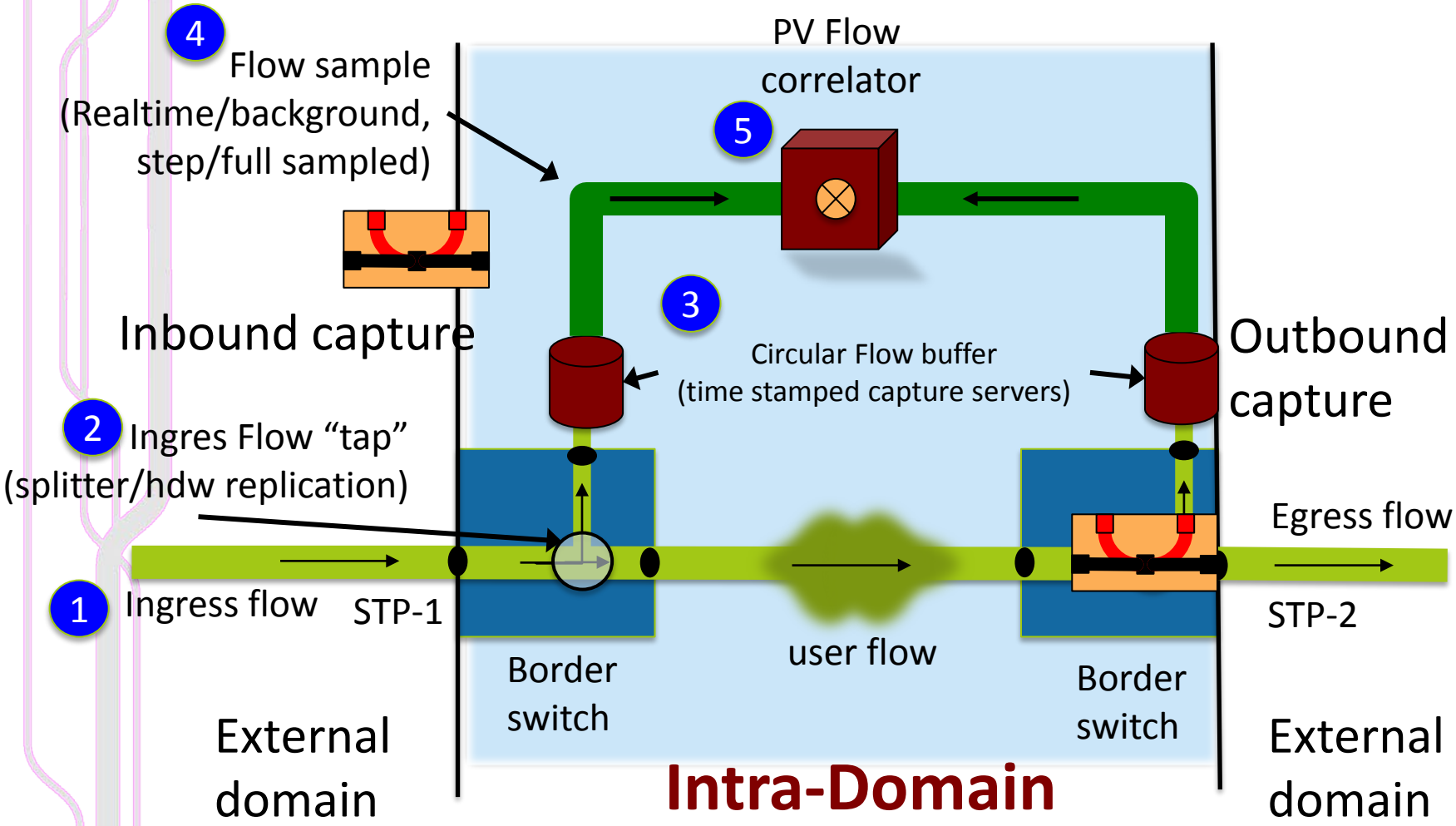
- Passive testing is preferred
- Passive PV is determined based upon observed traffic…
  - An independent source could send artificial (and potentially non-representative) traffic profiles
  - Or the test points can observe actual user traffic
- Does not require taking circuit out of service
  - The ingress test point and egress test points just listen (and capture)
- Can be run at any time(!) without affecting the user's actual flow(!!)
  - Provider can periodically perform a passive flow correlation to verify performance while in service
  - Provider can continuously monitor/capture flows in order to observe or replay specific failure events

    (Probably unrealistic to record large flows over long periods – but circular buffers can provide "black box" forensics.)

Active Performance Verification

- The active PV Server must be able to generate appropriate test flows:
  - Capacity - 100 gbps
  - Accurate shaping capabilities
- Some aspects of performance guarantees cannot be verified except in the presence of resource contention (competing congestive flows)
  - PV server(s) must be able to source multiple flows (at high capacity and with accurate shaping)
  - Servers must be able to coordinated/synchronize flows to create exterior conditions that meet particular verification requirements

A "Performance Flow Correlator"



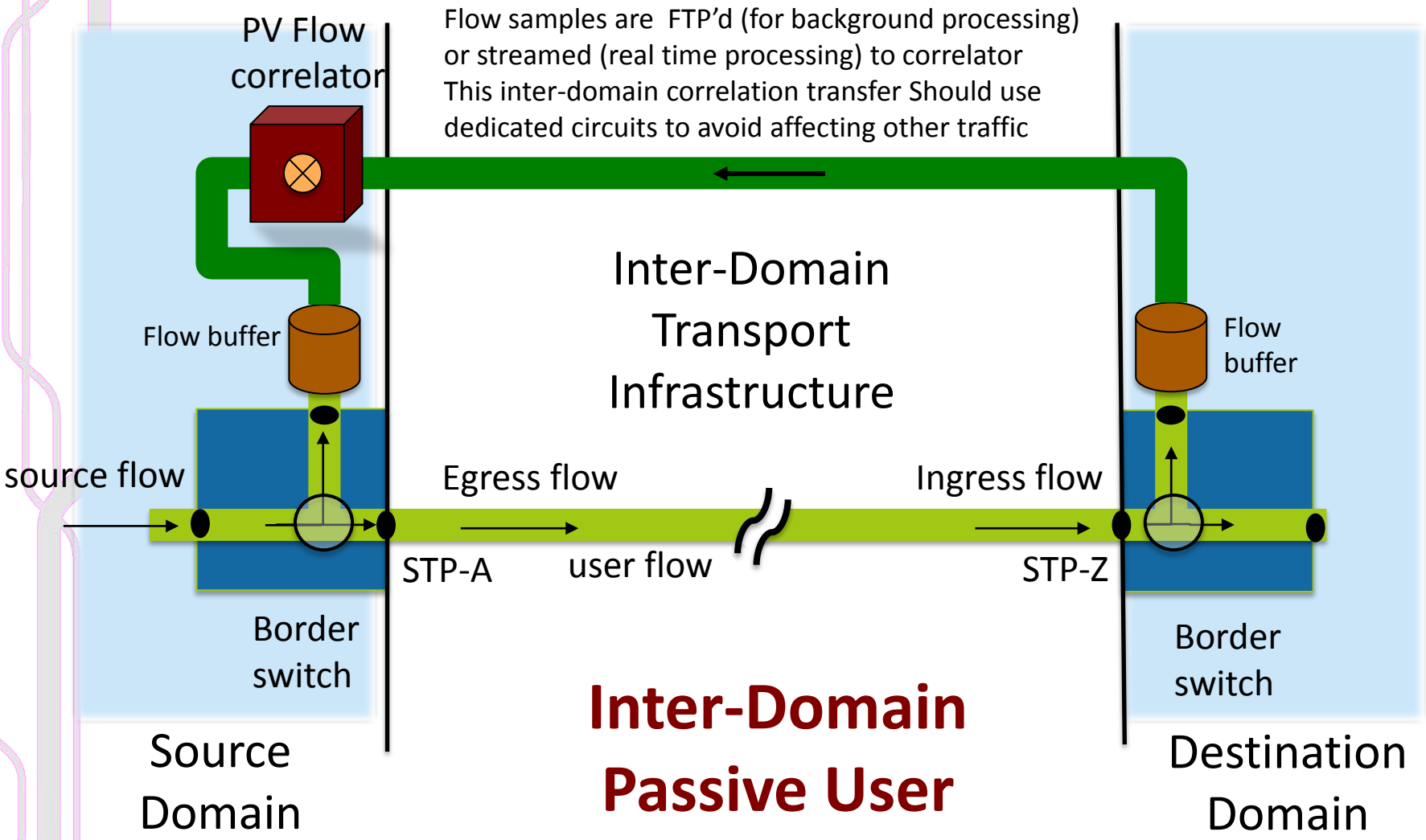
Intra-Domain Passive Provider PV Flow Correlation

NORDUnet
Nordic infrastructure for Research & Education

How the Flow Correlator Works

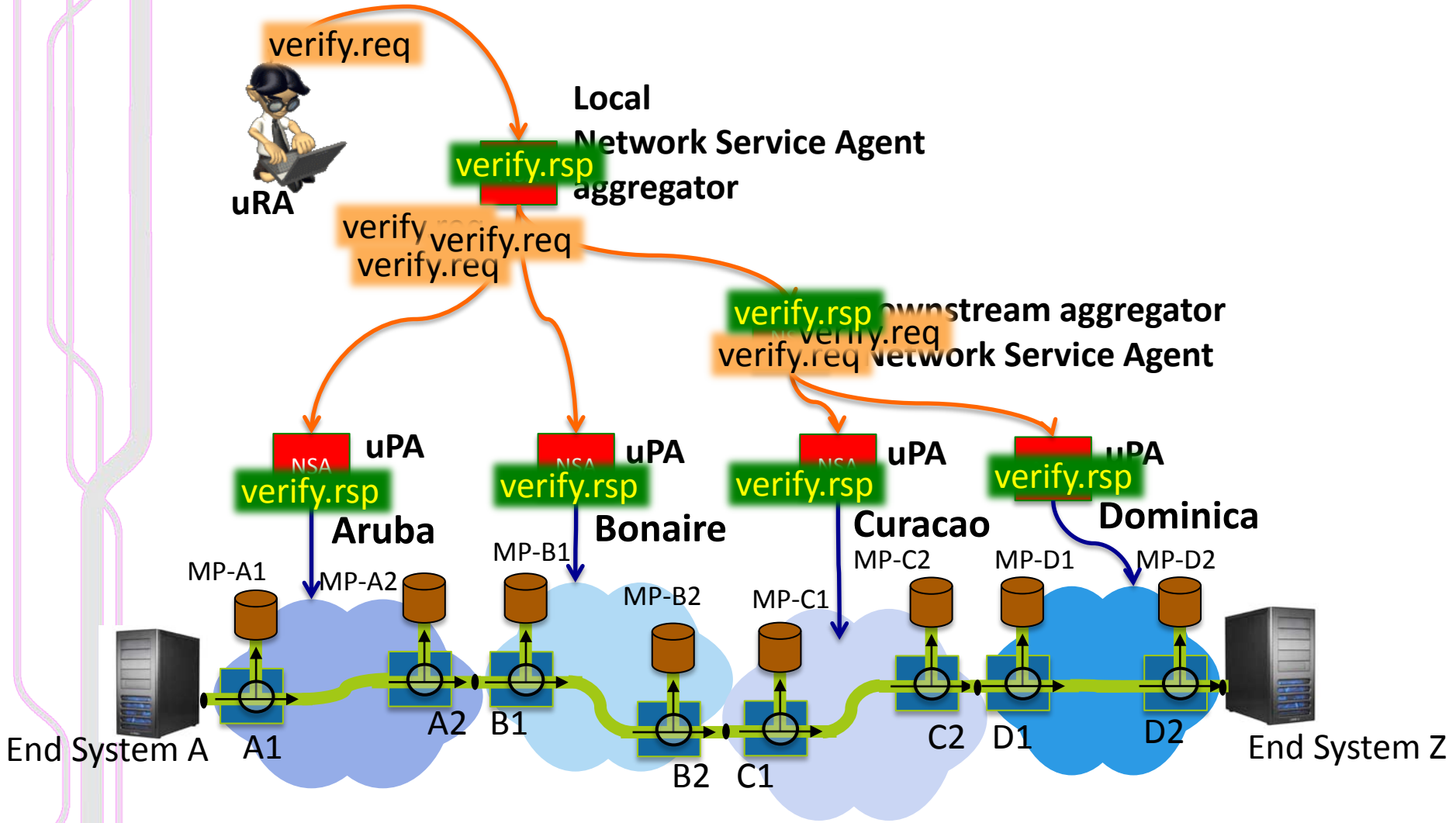
- The “PV tap” is implemented at every domain boundary interface
- The tap, when enabled, leads directly to a local capture device.
- The capture device must be able to:
  - Timestamp each datagram (depending on protocol).
  - Spool the captured stream to long term storage at line rate.
- The PV Capture Server can be sized and configured to capture an entire flow, or it can be sized to sample flows according to some rule or policy.
  - The correlation can be done in real time if engineered to do so…
  - or the flow can be captured and stored for later background analysis. … a few seconds later, or a few days later.
  - Correlation can be performed periodically, using short samples or real flows

An inter-Domain "Flow Correlator"



NORDUnet
Nordic infrastructure for Research & Education

End to End PV as a Service



Summary

- PV for performance guaranteed services involves understanding
  - A) What the service purports to be able to do
  - B) What the service instance is guaranteed
  - C) what we can effectively test
  - New technologies:
    - Advanced service architectures
    - Advanced and well engineered network/data plane architectures
    - Advanced high resolution timing and interface technologies.
  - Use existing tools and packages where they can fit into the picture
- Use the GLIF PVA Task Force to define a “stake in the sand” from which to start.

- The End