ANSE and PhEDEx

Integrating Network-Awareness and Network-Management into PhEDEx

Vlad Lapadatescu & Tony Wildish

Introduction

Overview

- Advanced Network Services for Experiments
- (Short) PhEDEx intro

- Current development efforts w.r.t. circuits and PhEDEx
  - Where/how it can be integrated
  - Previous results

- NSI circuits, issues encountered and proposed solution

- Summary and future plans

ANSE

A project funded by NSF CC-NIE program

ANSE - Advanced Network Services for Experiments

Integrate network awareness into the software stacks of experiments

- PhEDEx for CMS
- Panda for ATLAS

Started Jan 2013

- Build on top of existing services (LHCOPN, LHCONE)

PIs:

- Harvey Newman, PI, Caltech
- Shawn McKee, co-PI, University of Michigan
- Paul Sheldon, co-PI, Vanderbilt University
- Kaushik De, co-PI, University of Texas at Arlington

PhEDEx Overview

The data management transfer tool for CMS (since 2004)

Loosely coupled set of agents written in Perl interacting via central DB

- **central agents** (ex. **FileRouter** agent)
- **site agents** running at various T1s and T2s (ex. **FileDownload** agent)
- each agent performs a independent single task

Common workflow:

- Front-end used to request data to sites
- Central agents compute paths of least cost, schedule transfers, etc
- Site agents execute transfer tasks

FileRouter (central) agent builds transfer queue per destination
FileDownload (site) agent examines its queue, processes it & reports back

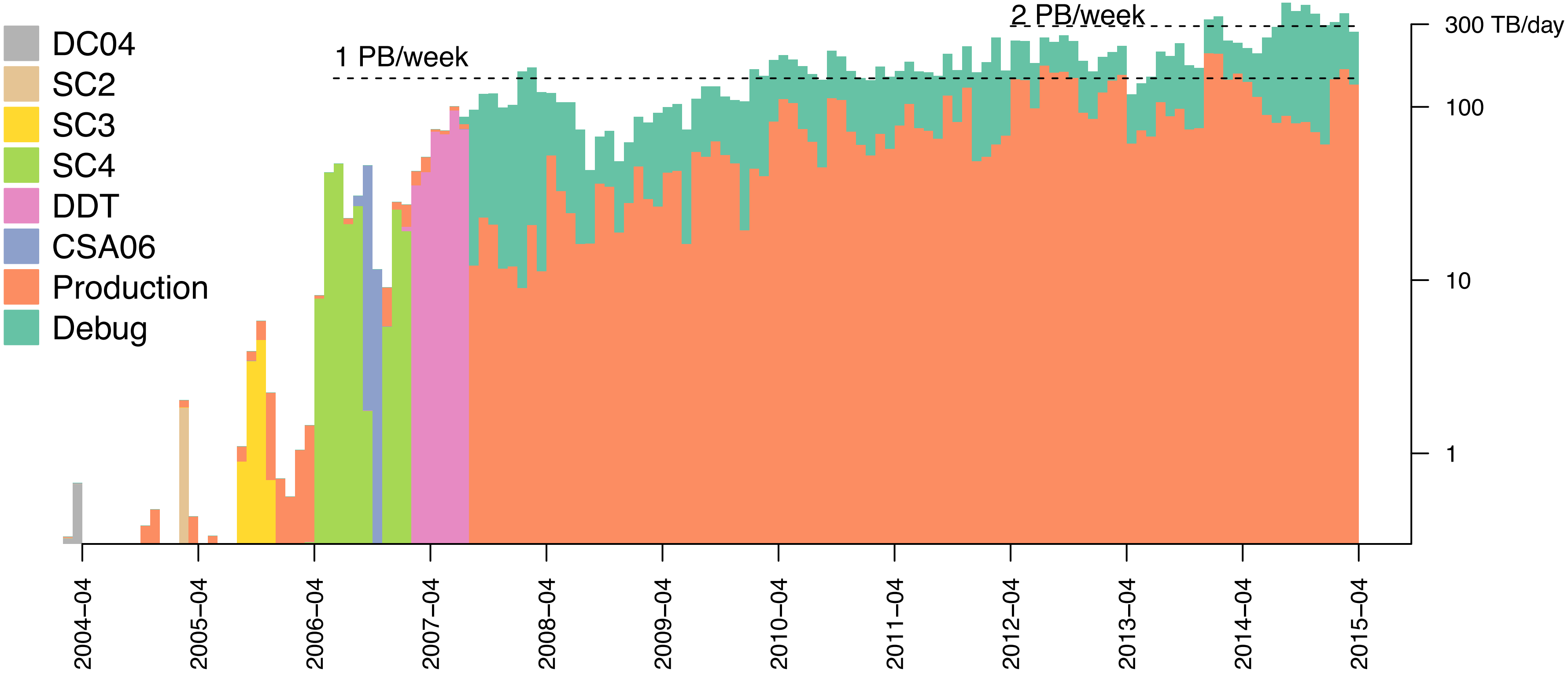
PhEDEx Overview 2

PhEDEx is:

- not necessarily "near" the storage (i.e. same subnet)
- high level software … only knows about:
  - datasets, blocks, files
  - Hostnames/IPs from URLs
  - Path of files

When transferring data, PhEDEx provides…

- Datasets, blocks & file names & sizes
- SURL (storage farm hostname, local file path)
- Information about transfer queues
- Limited monitoring information (src-dst rates, quality)


DC04
SC2
SC3
SC4
DDT
CSA06
Production
Debug
1 PB/week
2 PB/week
300 TB/day
100
10
1
2004–04
2005–04
2006–04
2007–04
2008–04
2009–04
2010–04
2011–04
2012–04
2013–04
2014–04
2015–04

ANSE & PhEDEx

Goals:

- Enhance PhEDEx with circuit awareness capabilities
- Provide a tool which can be used by others (CMS, non-CMS, non-HEP)
- Enhance PhEDEx with knowledge of network status (not covered here)

Motivation*:

- More deterministic transfers (schedule jobs with data, set/meet deadlines)
- Data prioritization over other traffic

PhEDEx integration possibilities:

- In the FileDownload agent (site level):
  - + Compromise between desired functionality and complexity
  - - Only has a local view

- In the FileRouter agent (central level):
  - + Has a global view of the whole system
  - - Harder to implement and optimize

* Provided that guaranteed BW is available

Initial prototype

Modified the FileDownload agent to:

- Examine its own download queue
- Determine whether a circuit is useful (expected vs. achieved transfer rate)
- Request a circuit (using DYNES)
  - If circuit was established:
    - convert transfer URLs to use the new L3 path
    - start new transfer using the updated URLs
- Manage the lifecycle of the circuit

+ No modifications to PhEDEx DB

+ All control logic in the FileDownload agent

+ Transparent for all other network users

− Relied on FDT as a transfer backend – not widely used

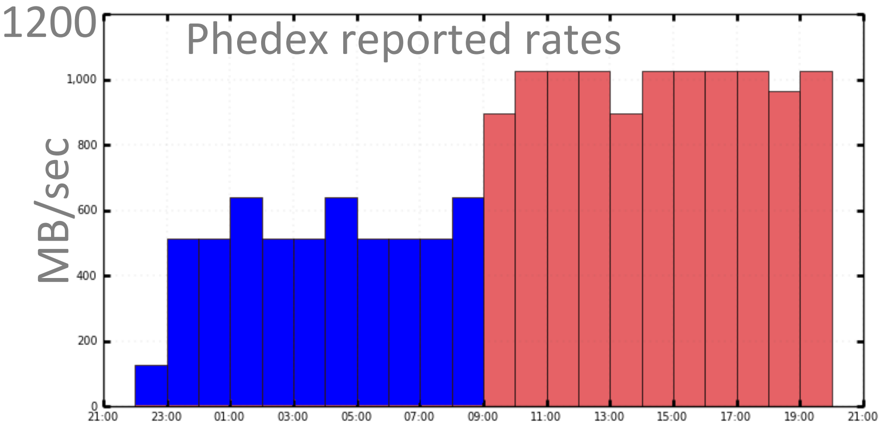
− Embedded in the FileDownload agent, could not be used by external apps

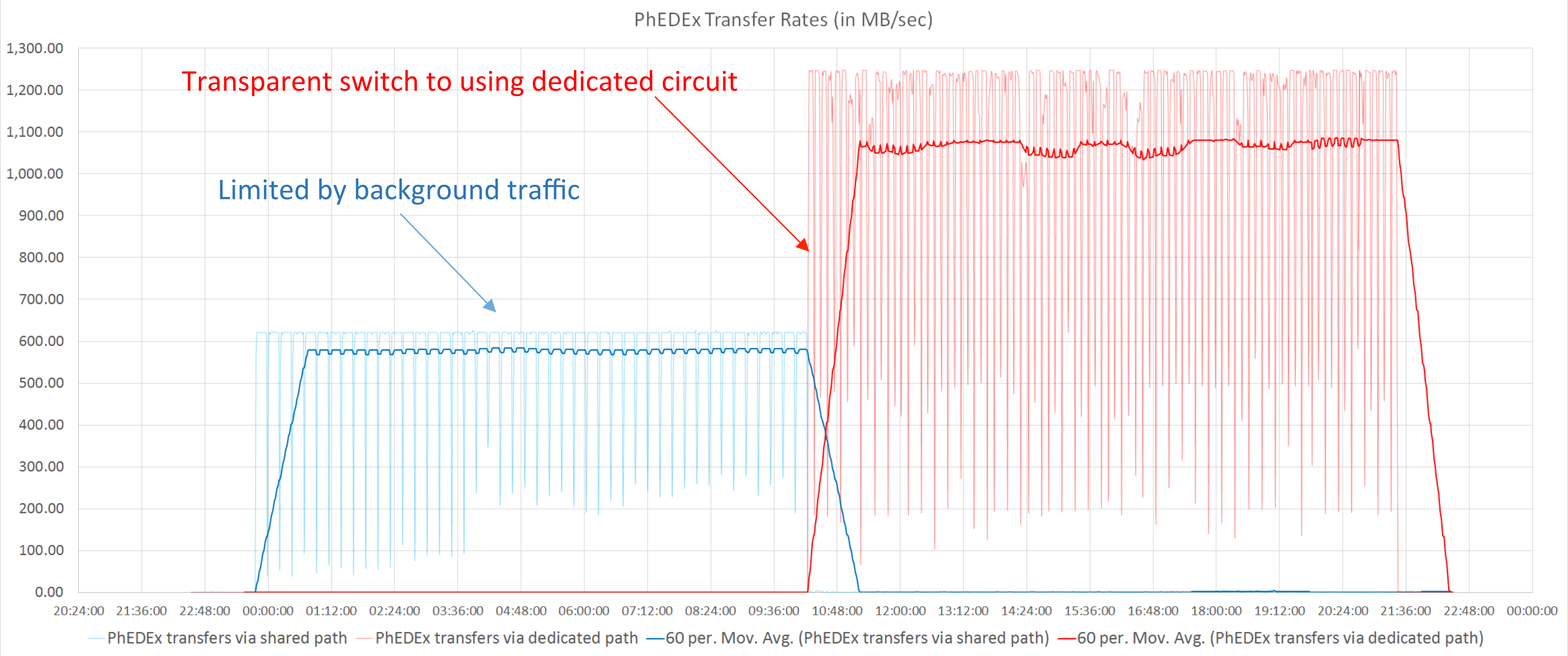
− Single-purpose, could not be extended to use other circuit providers

− DYNES no longer supported, not a candidate for production use

Results using the prototype

- **Seamless path switch**
- Per job link rates with PhEDEx traffic
  - ~620MB/sec -> 1060 to 1250MB/sec
- Average link rates with PhEDEx traffic
  - ~570MB/sec -> ~1050MB/sec


Phedex reported rates


PhEDEx Transfer Rates (in MB/sec)

Transparent switch to using dedicated circuit

Limited by background traffic

Using NSI to create circuits

'Network Service Interface'

- NSI is an advance-reservation based protocol
- Supports tree and chain model of service chaining

Two phase reservation system

- First phase: availability is checked, if available, resources are held
- Second phase:
  - the requester either commits or aborts a held reservation
  - should the requester fail, reservation expires after a timeout

NSI reservation properties

- Source, destination endpoints (mandatory)
- Start time, end time, reserved bandwidth (optional)

Limitations

- Only supplies a L2 circuit
- Circuit ends at site border router, not at storage farm nodes
- Some providers don't guarantee BW
- Not widely adopted - yet

Issues in dealing with L2 circuits

Transfer backends can't directly use the NSI L2 circuit

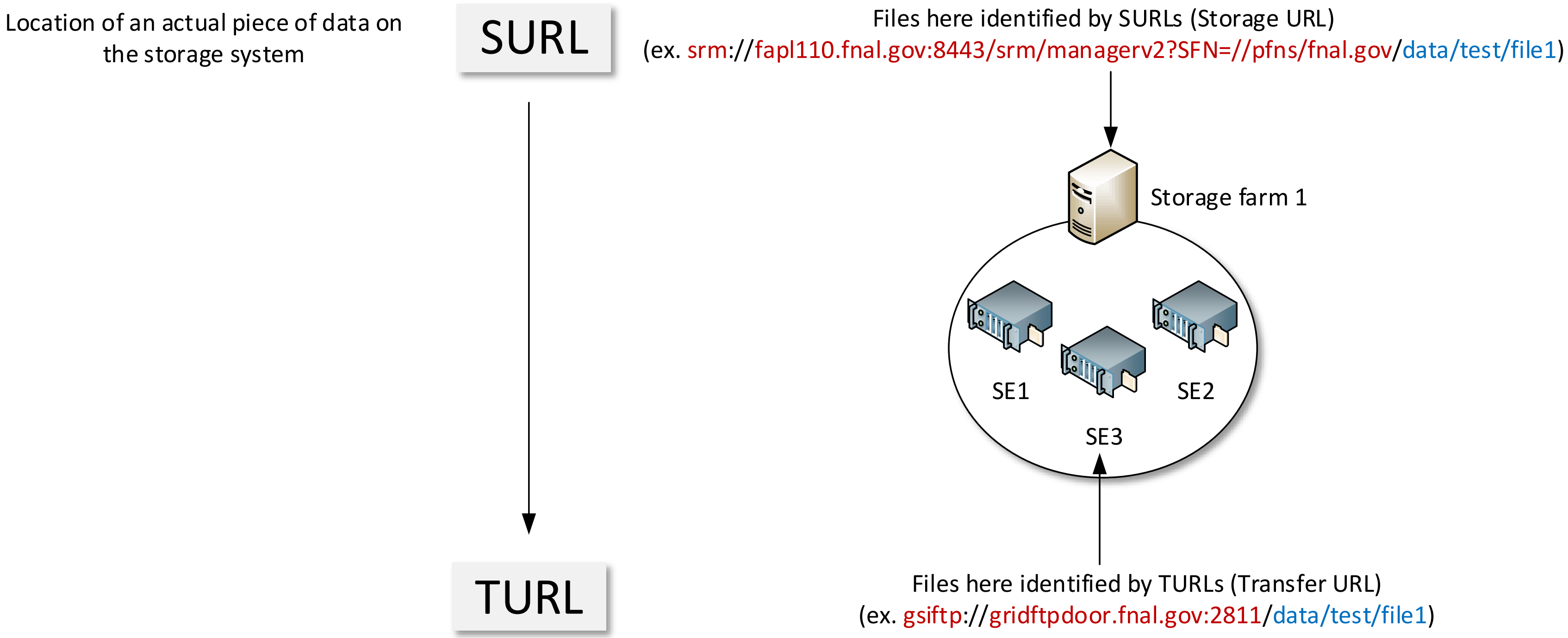
Establishing L3 path to storage requires:

- Some topology knowledge
- Routing information
- Direct access to the site's network equipment
- => non-trivial to establish!

PhEDEx is a very high-level software -> Can only provide

- Datasets, blocks & file names and sizes
- SURL (Storage URL)
  - Storage farm hostname
  - Local file path

SURLs and TURLs…



Location of an actual piece of data on the storage system

SURL

Files here identified by SURLs (Storage URL)
(ex. srm://fapl110.fnal.gov:8443/srm/managerv2?SFN=//pfns/fnal.gov/data/test/file1)

TURL

Files here identified by TURLs (Transfer URL)
(ex. gsiftp://gridftpdoor.fnal.gov:2811/data/test/file1)

SURLs to TURLs (FTS & SRM)

- Get source TURL (call srmPrepareToGet)
- Get destination TURL (call srmPrepareToPut)
- Assuming that the TURL-s are gridftp endpoints, start gridftp copy
- Monitor transfer progress
- Release TURLs

So what do we do?

Technical constraints:

- Only a L2 circuit
- L2 circuit ends in the site's border router
- Limited feedback in case of errors
- NSI adoption in production is still limited

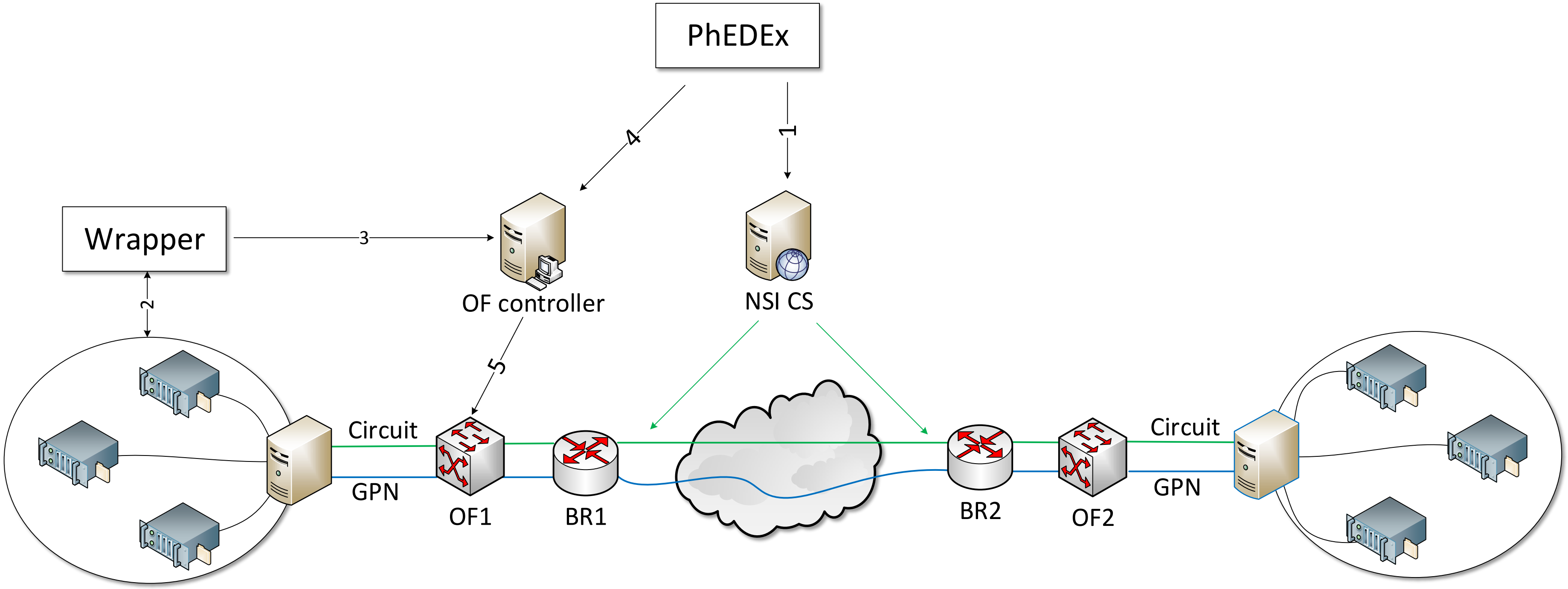
All solutions of creating a L3 path rely either on

- privileged access on site's servers/routers
- specialised hardware in place (OF capable)

Our solution must:

- deal with sites serving multiple VOs
- potentially deal with privileged & non privileged traffic on the same path
- work with FTS/SRM/gridFTP
- be as non-intrusive into sites operations as possible

Proposed solution diagram



1. Request circuit between site A and site B
2. Wrapper gets IPs of all servers involved in the transfer
3. Wrapper passes this information to the OF controller
4. PhEDEx informs the OF controller that a circuit exists between the two sites
5. OF controller adds routing info in the OF switches that direct traffic on the subnet to the circuit

Least intrusive option (least privilege), but requires OF controllers in the network

Summary & future plans

PhEDEx is ready to use circuits in production as soon as they are available

- No modifications to the PhEDEx DB
- Control logic is in the FileDownload agent
- Lifecycle handled by the ResourceManager
- Transparent for all other PhEDEx instances

ResourceManager can be used as a 3rd party tool

- No CMS-specific parts

Future plans:

- Solve the issue of how to route data once a circuit is active
- Demonstrate circuit management capabilities between select sites
- Demonstrate improvement while using circuits

Questions?