Technical Tests for LHCONE & Named Data Networking update

Duncan Rand and Simon Fayer

Technical Tests for LHCONE

Duncan Rand and Simon Fayer

GridPP
UK Computing for Particle Physics

Imperial College London

- Many WLCG sites use LHCONE
- GridPP would like to understand better the procedure for UK sites joining LHCONE
- Imperial College volunteered as a test case a while back
- Roll out of Janet's new London network including router upgrade was completed this summer
- This enabling us to proceed in earnest, started early August 2015

LHCONE VRF and VLAN

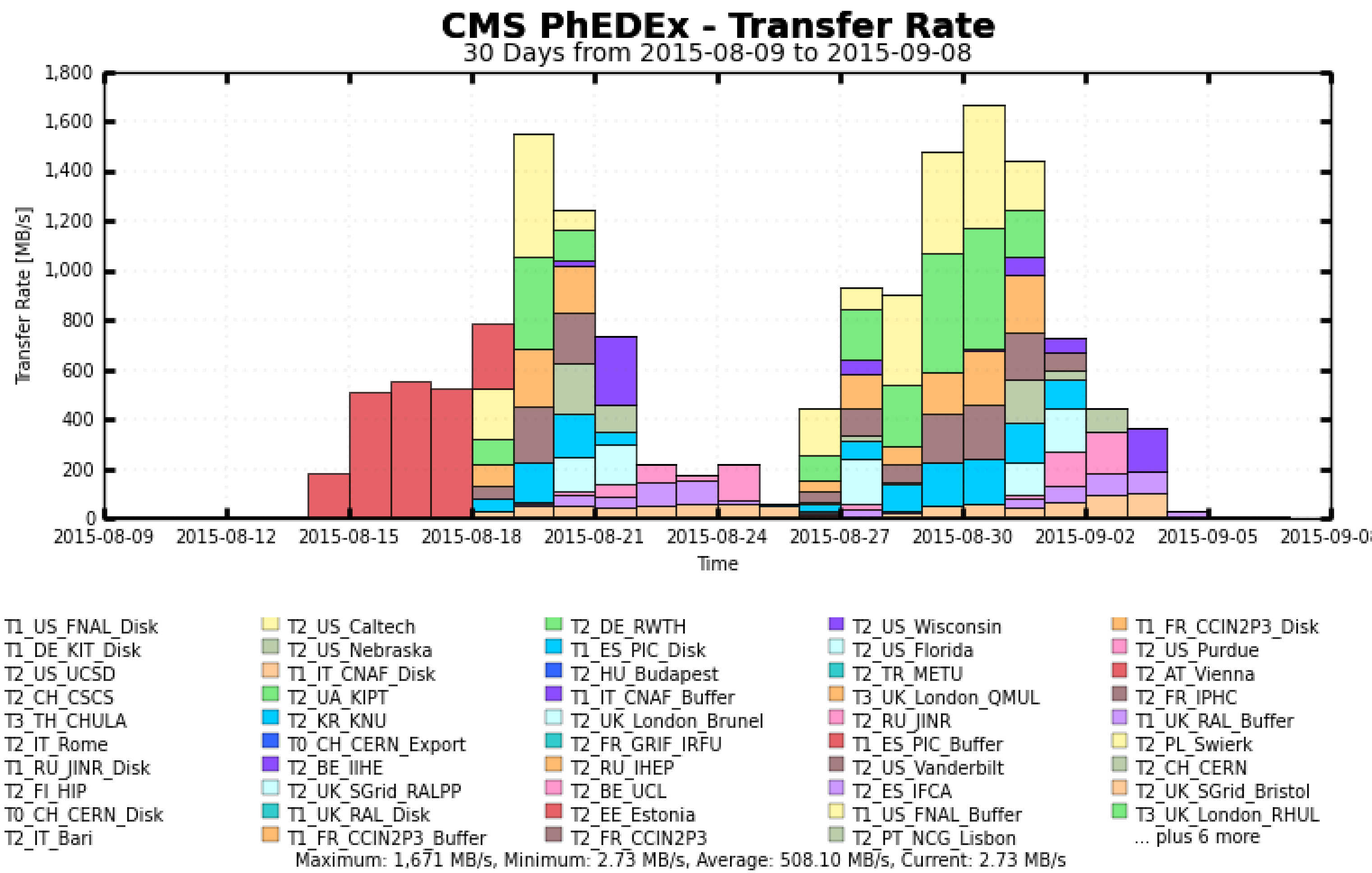
- Janet set up an LHCONE VRF (virtual routing and forwarding, Wikipedia: 'allows multiple instances of a routing table within the same router at the same time')
- Initially Imperial network team set up a new VLAN on the LHCONE VRF with two new subnets for testing
- We signed the LHCONE AUP
- Not easy for us to move hosts across, so we requested adding the LHCONE VRF into our existing grid subnets
- Solution chosen involved a downstream device to maintain routing adjacencies to both default and LHCONE VRFs
- Dynamic routing set up using the new spare subnets
- We moved one host in and checked accessibility of LHCONE and general internet connectivity
- That worked OK so whole grid subnet moved across to this routing arrangement

Debugging issues

- IPv6 connectivity was fine then stopped working. Router advertisements were missing the prefix & MTU information sections compared to the other IPv6 subnets – soon fixed
- Then we detected a loss of connectivity to CERN VOMS server: there was a problem propagating our routes into LHCONE – soon fixed
- Next we detected an asymmetric routing problem with FNAL: packets sent over LHCONE were coming back over the general internet
- FNAL were contacted (via Janet) and updated their routing configuration manually
- So now everything seemed to be working OK and we could start load testing
- Many thanks to Phil Myers and the team from Janet (Rob Evans & Dave Tinkler) for their hard work

LHCONE Load Testing

- Made use of CMS load test infrastructure
- Copy data sets out of IC to other CMS sites known to be on LHCONE using CMS debug instance of PhEDEx
- Set suitable rate to every destination site



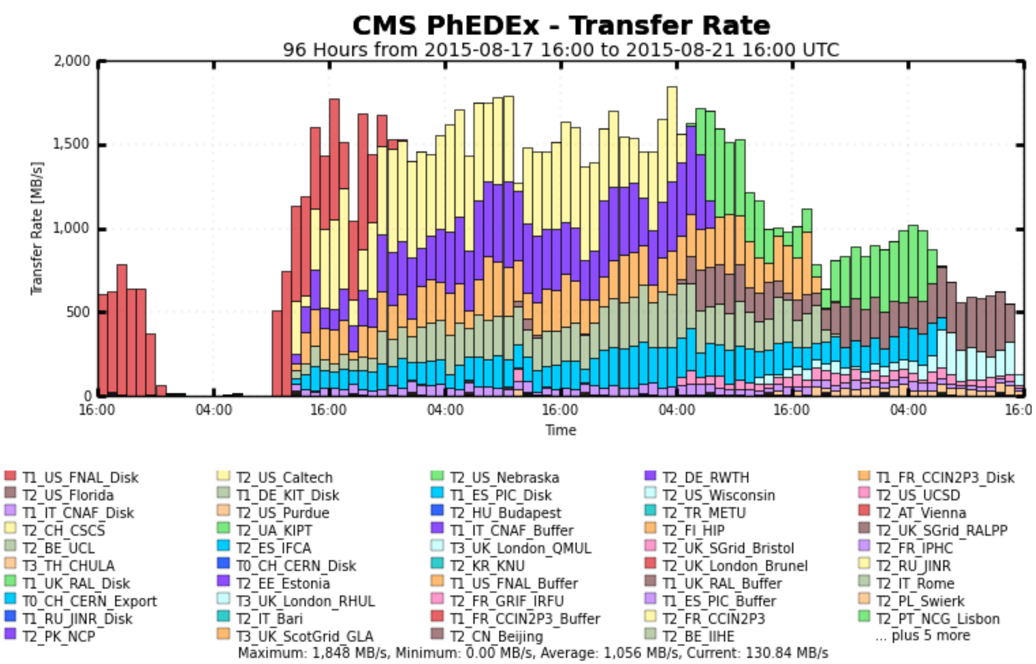
GridPP
UK Computing for Particle Physics

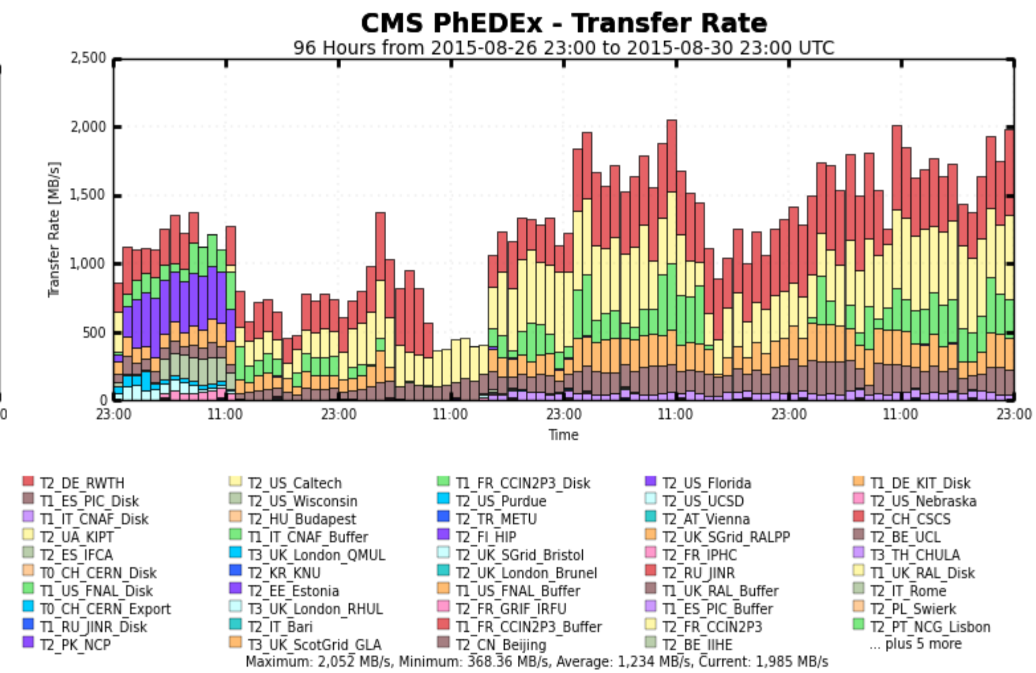
Imperial College London

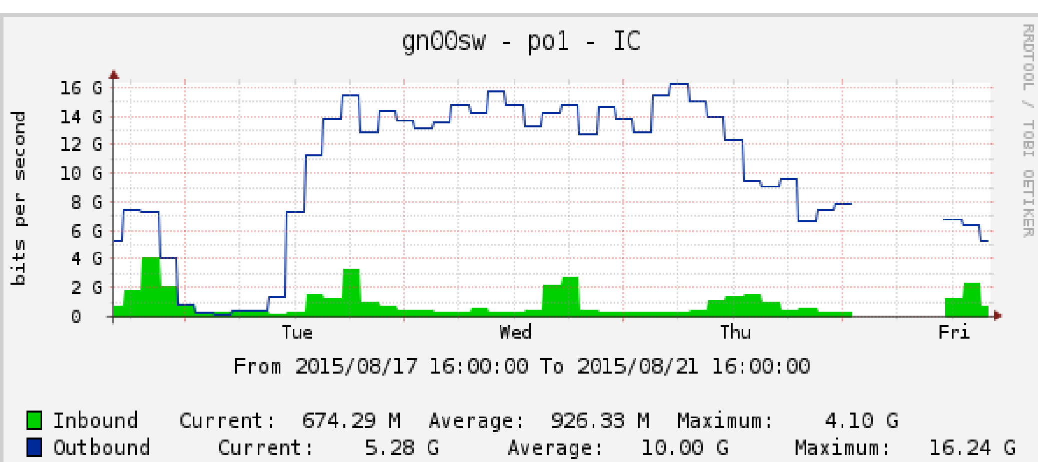
Outbound

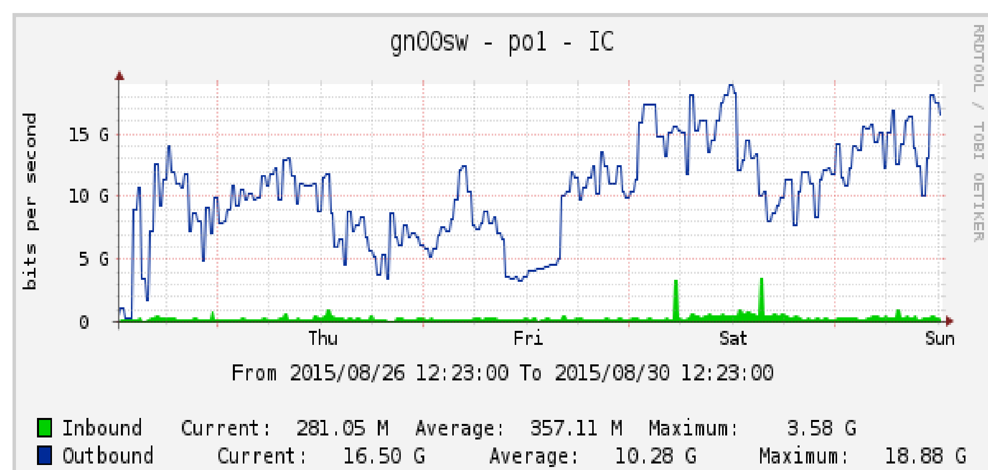
Before

After





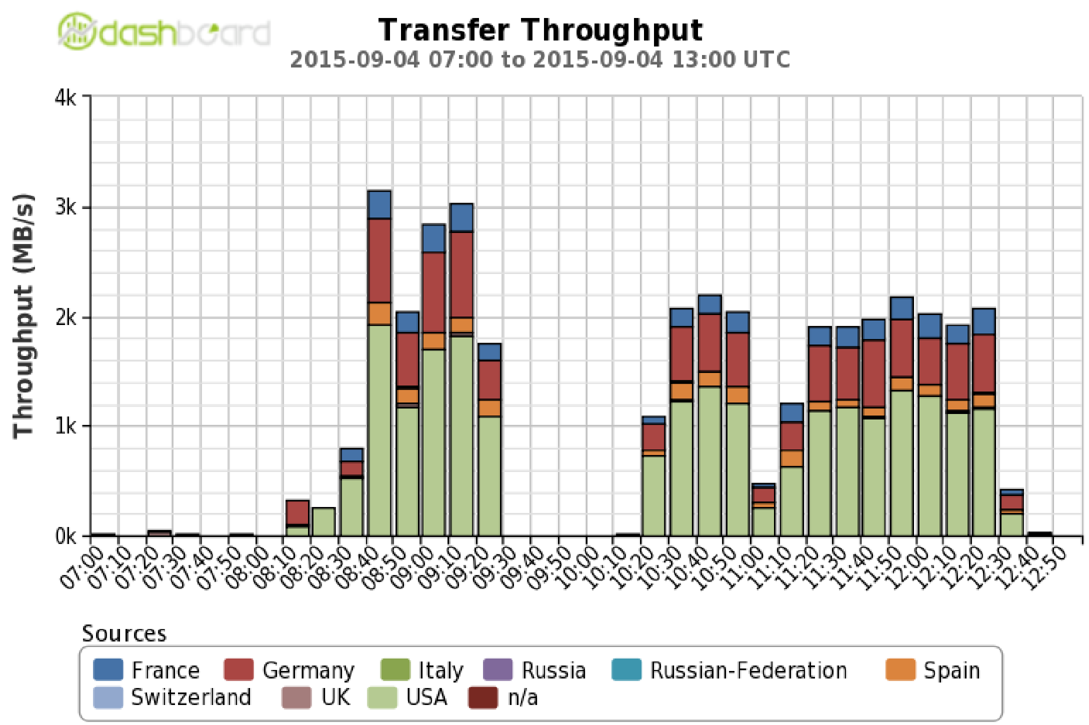


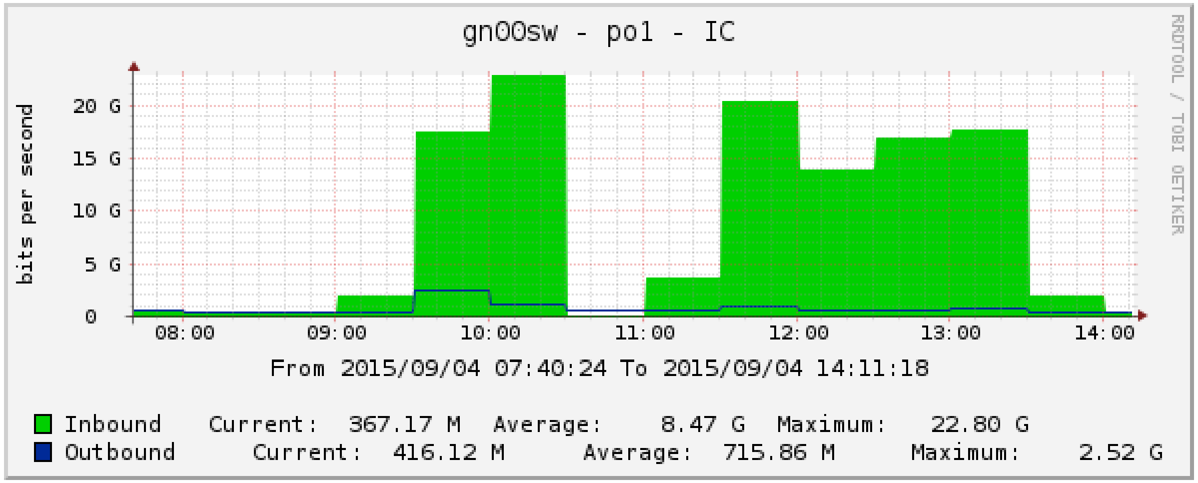


Inbound

- Also did inbound tests
- Found a maximum rate of about 26 Gbps and that the two links were not well balanced (one link was saturating at 20 Gbps, the other 6Gbps)
- Rebalanced by moving some storage pool nodes from one subnet to the other

- Long term solution is to configure Equal Cost Multi Path ('ECMP')

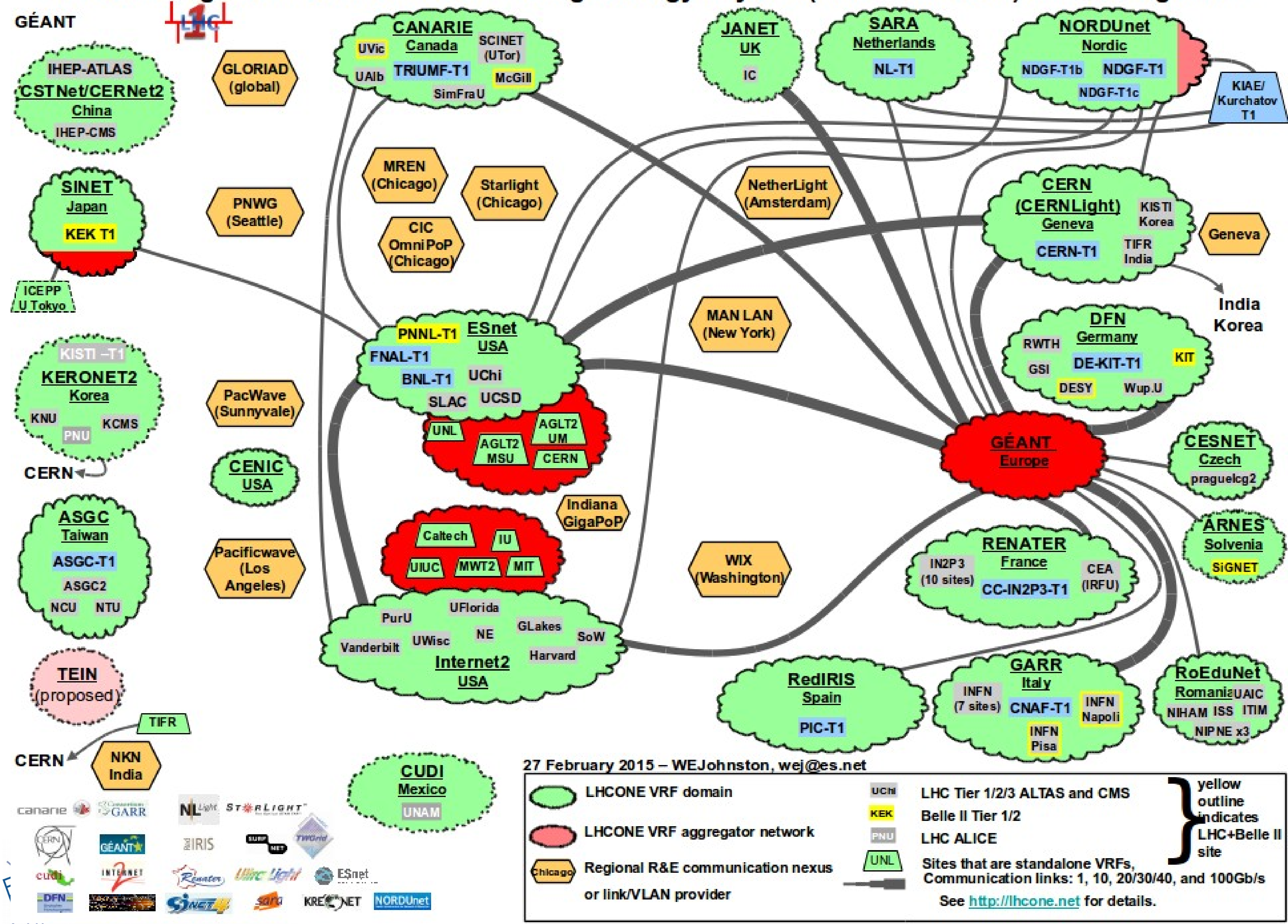




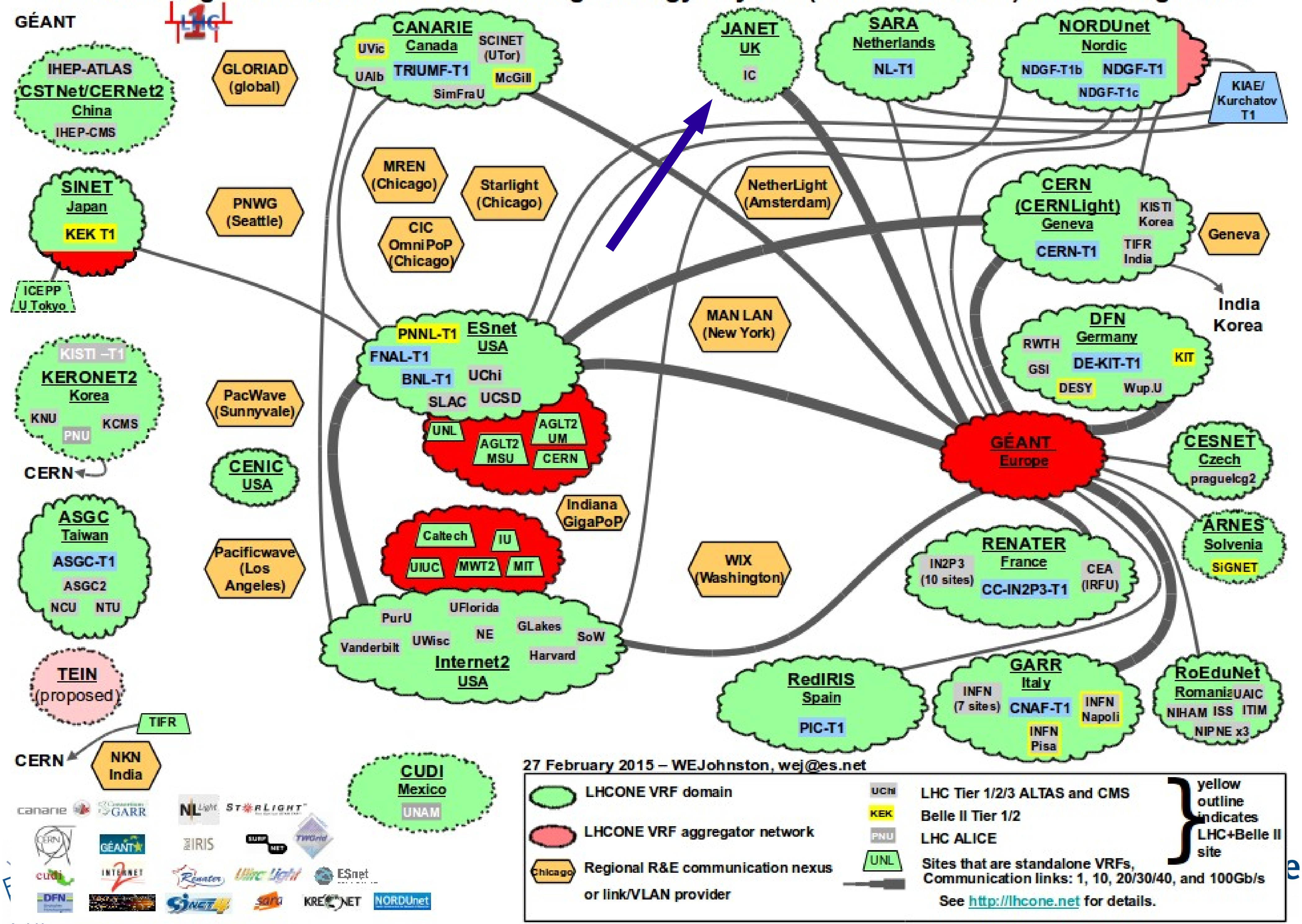
GridPP UK Computing for Particle Physics

Imperial College London

LHCONE: A global infrastructure for the High Energy Physics (LHC and Belle II) data management



LHCONE: A global infrastructure for the High Energy Physics (LHC and Belle II) data management



Named Data Networking update

Duncan Rand and Simon Fayer

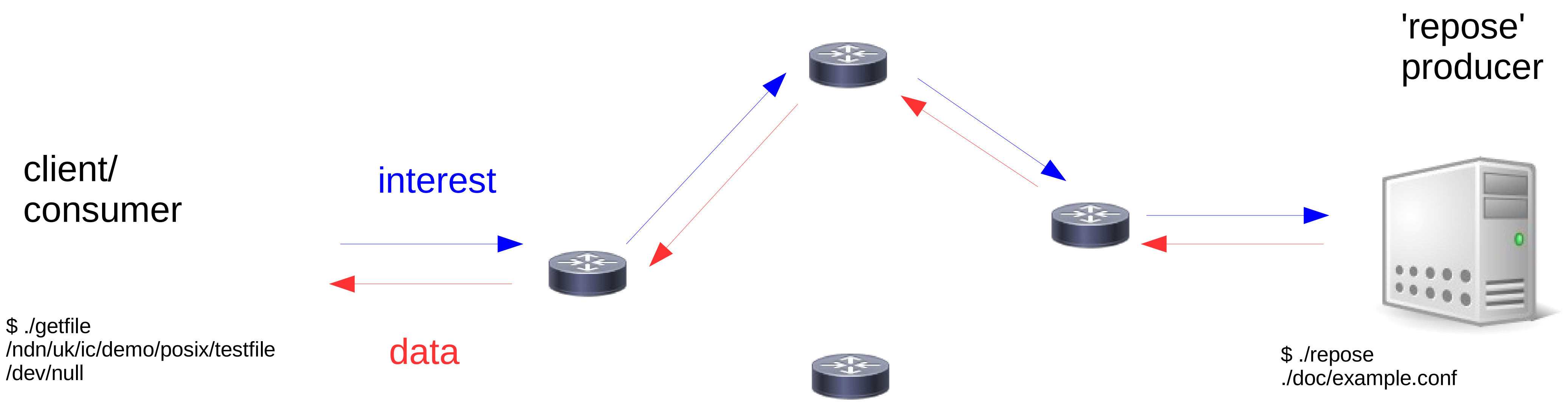
GridPP UK Computing for Particle Physics

Imperial College London

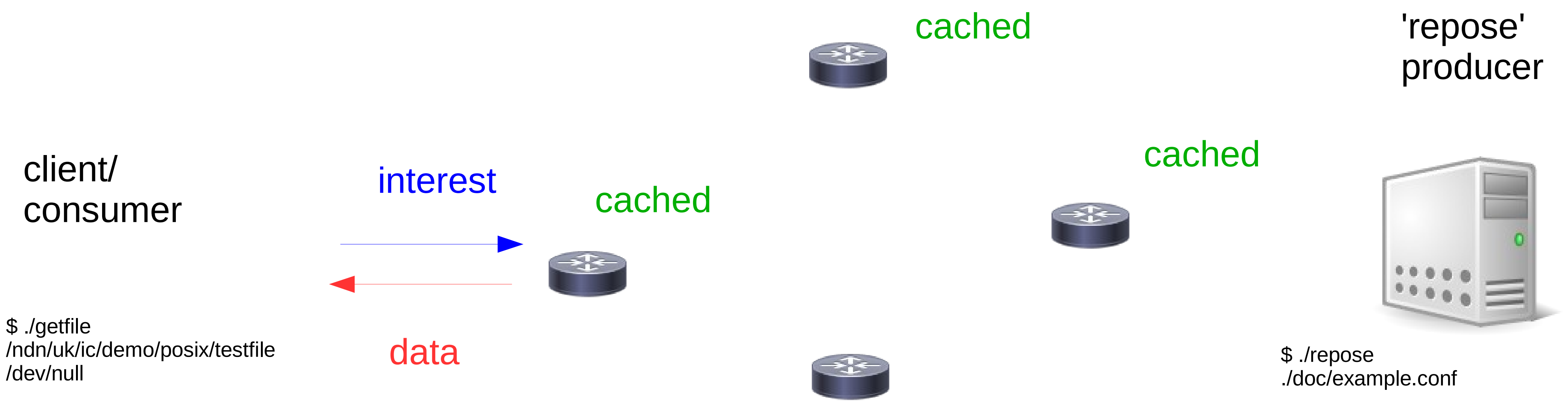
Named Data Networking

- Named Data Networking (NDN) is an instance of the Information Centric Networking (ICN) research field
- NDN is a novel NSF-funded internet architecture in which data is named
  - e.g. /ndn/uk/ic/demo/posix/testfile4
- Data packets are also cryptographically signed to ensure provenance and integrity
- The naming and signing of data means that it loses its dependency on location; it is no longer necessary to know which host to go to to get a particular file
- This also means that data can now be cached, for example in network routers themselves
- So now, rather than connecting to a server known to have a file, getting a file involves a request to the named data network for that file
- The request is in the form of an 'interest packet' containing the name of the data sought
- On receipt of an interest packet network routers either serve the data segment if it is in their cache or if not they forward the interest packet towards a copy of the data
- The caching of data should improve network efficiency

First request for file /ndn/uk/ic/demo/posix/testfile



Second request for file /ndn/uk/ic/demo/posix/testfile



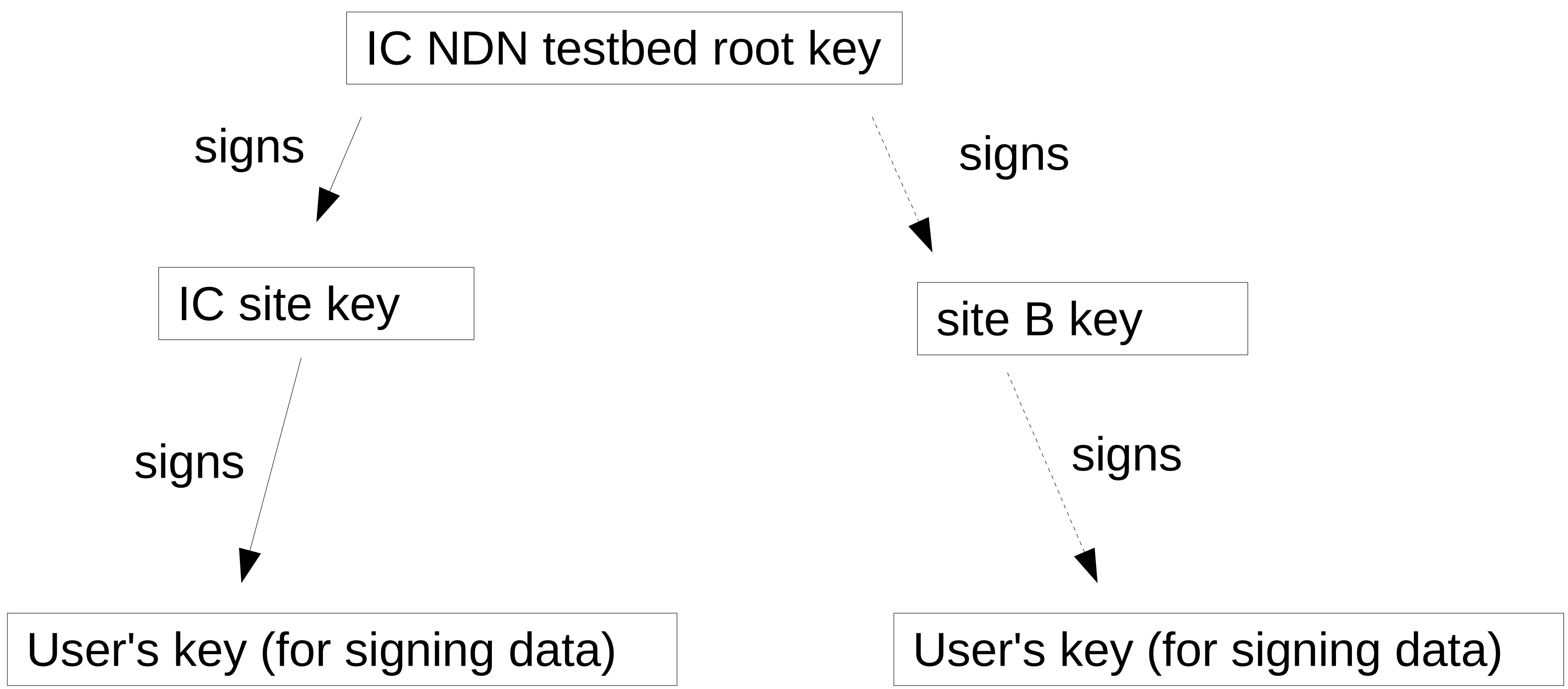
- On second read the file is cached in the closest NDN router content store (memory)
- Results in much faster response and reduces network usage
- File is also cached in more distant routers, perhaps usable by other nearby consumers
- Need to check signature validity to ensure data provenance, integrity etc
- Note, public keys used to verify signatures can themselves be retrieved via NDN

GridPP UK Computing for Particle Physics

Imperial College London

Key trust model in NDN

- NDN applications use a simple certification chain trust model
- Imperial NDN testbed root key acts as trust anchor



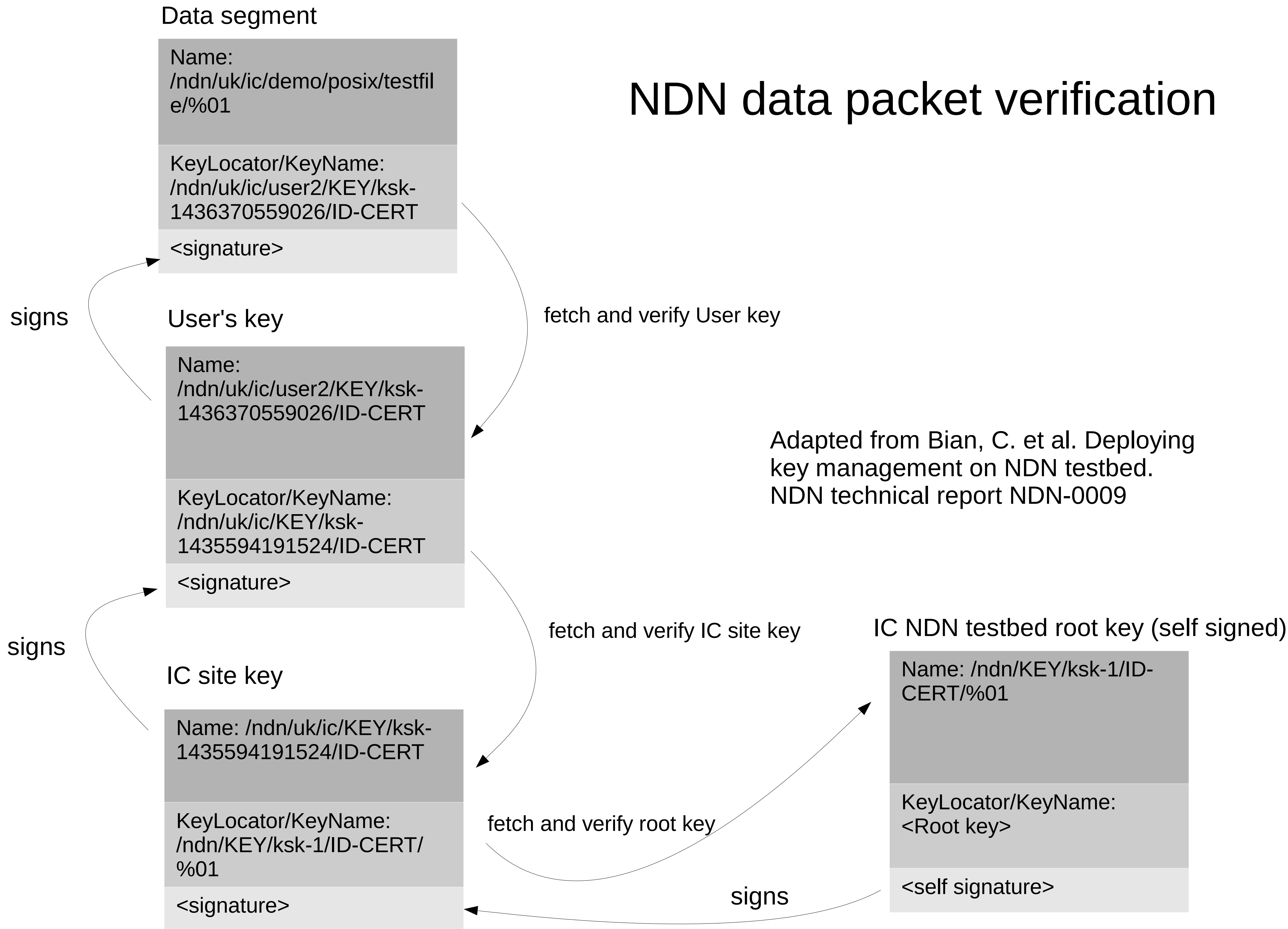
GridPP UK Computing for Particle Physics

Imperial College London

Key trust model

Entity	Identity name	Example	Certificate name example
root	/network	/ndn	/ndn/KEY/ksk-1/ID-CERT/%01
site	/<network>/<site>	/ndn/uk/ic	/ndn/uk/ic/KEY/ksk-2/ID-CERT/%01
user	/<network>/<site>/<user-name>	/ndn/uk/ic/user2	/ndn/uk/ic/user2/KEY/ksk-3/ID-CERT/%01

- Each data packet contains a KeyLocator/KeyName field
- This gives the name of the key used to sign the packet
- The application can then request that certificate as another NDN data segment
- This is done recursively until the locally available trust anchor (root key) is reached


NDN data packet verification
Data segment
Name: /ndn/uk/ic/demo/posix/testfile/%01
KeyLocator/KeyName: /ndn/uk/ic/user2/KEY/ksk-1436370559026/ID-CERT
<signature>
signs
fetch and verify User key
User's key
Name: /ndn/uk/ic/user2/KEY/ksk-1436370559026/ID-CERT
KeyLocator/KeyName: /ndn/uk/ic/KEY/ksk-1435594191524/ID-CERT
<signature>
signs
fetch and verify IC site key
IC site key
Name: /ndn/uk/ic/KEY/ksk-1435594191524/ID-CERT
KeyLocator/KeyName: /ndn/KEY/ksk-1/ID-CERT/%01
<signature>
fetch and verify root key
IC NDN testbed root key (self signed)
Name: /ndn/KEY/ksk-1/ID-CERT/%01
KeyLocator/KeyName: <Root key>
<self signature>
signs
Adapted from Bian, C. et al. Deploying key management on NDN testbed. NDN technical report NDN-0009

Validator configuration file

Consists of rules and anchors

```
$ more validator.conf
rule
{
  id "Test Validation Rule"
  for data
  filter
  {
    type name
    name /ndn/uk/ic
    relation is-prefix-of
  }
  checker
  {
    type customized
    sig-type rsa-sha256
    key-locator
    {
      type name
      name /ndn/uk/ic/user2/KEY/ksk-1436370559026/ID-CERT
      relation equal
    }
  }
}
trust-anchor
{
  type file
  file-name /etc/ndn/keys/root.cert
}
```

http://named-data.net/doc/ndn-cxx/current/tutorials/security-validator-config.html

Validator configuration file

```
$ more validator.conf
rule
{
  id "Test Validation Rule"
  for data
  filter
  {
    type name
    name /ndn/uk/ic
    relation is-prefix-of
  }
  checker
  {
    type customized
    sig-type rsa-sha256
    key-locator
    {
      type name
      name /ndn/uk/ic/user2/KEY/ksk-1436370559026/ID-CERT
      relation equal
    }
  }
}
trust-anchor
{
  type file
  file-name /etc/ndn/keys/root.cert
}
```

Rule matches data packets with name that has prefix /ndn/uk/ic

http://named-data.net/doc/ndn-cxx/current/tutorials/security-validator-config.html

GridPP UK Computing for Particle Physics

Imperial College London

Validator configuration file

```
$ more validator.conf
rule
{
  id "Test Validation Rule"
  for data
  filter
  {
    type name
    name /ndn/uk/ic
    relation is-prefix-of
  }
  checker
  {
    type customized
    sig-type rsa-sha256
    key-locator
    {
      type name
      name /ndn/uk/ic/user2/KEY/ksk-1436370559026/ID-CERT
      relation equal
    }
  }
}
trust-anchor
{
  type file
  file-name /etc/ndn/keys/root.cert
}
```

Checker requires that key must have signature type rsa-sha256 with certificate name /ndn/uk/ic/user2/KEY/ksk-1436370559026/ID-CERT

http://named-data.net/doc/ndn-cxx/current/tutorials/security-validator-config.html

Validator configuration file

```
$ more validator.conf
rule
{
  id "Test Validation Rule"
  for data
  filter
  {
    type name
    name /ndn/uk/ic
    relation is-prefix-of
  }
  checker
  {
    type customized
    sig-type rsa-sha256
    key-locator
    {
      type name
      name /ndn/uk/ic/user2/KEY/ksk-1436370559026/ID-CERT
      relation equal
    }
  }
}
trust-anchor
{
  type file
  file-name /etc/ndn/keys/root.cert
}
```

Trust anchor tells validator which certificates are valid immediately.

http://named-data.net/doc/ndn-cxx/current/tutorials/security-validator-config.html

NLSR – Named data Link State Routing Protocol

- Previously routes to names between hosts in the testbed were added manually, e.g.

  nfdc register /ndn/uk/ic udp4://123.123.123.123

- NLSR protocol propagates reachability to names (rather than IP prefixes) analogous to BGP
- Uses NDN interest and data packets
- We have now installed NLSR on our local testbed
- NLSR uses same/similar trust model as described above

Key trust model in NLSR

- Certification chain trust model

```
                               root
                                |
                 +--------------+---------------+
               site1                          site2
                 |                              |
       +---------+---------+                    +
   operator1           operator2            operator3
       |                   |                    |
  +-----+-----+       +----+-----+       +-----+-----+-----------+
router1    router2  router3    router4  router5    router6  router7
  |          |        |          |        |          |        |
  +          +        +          +        +          +        +
NLSR       NSLR      NSLR       NSLR     NSLR       NSLR     NSLR
```

Entity	Identity name	Example	Certificate name example
root	/network	/ndn	/ndn/KEY/ksk-1/ID-CERT/%01
site	/<network>/<site>	/ndn/uk/ic	/ndn/uk/ic/KEY/ksk-2/ID-CERT/%01
operator	/<network>/<site>/%C1.Operator/<operator-name>	/ndn/uk/ic/%C1.Operator/ndnops	/ndn/uk/ic/%C1.Operator/ndnops/KEY/ksk-3/ID-CERT/%01
router	/<network>/<site>/%C1.Router/<router-name>	/ndn/uk/ic/%C1.Router/router1	/ndn/uk/ic/%C1.Router/router1/KEY/ksk-4/ID-CERT/%01
NLSR	/<network>/<site>/%C1.Router/<router-name>/NLSR	/ndn/uk/ic/%C1.Router/rt1/NLSR	/ndn/uk/ic/%C1.Router/rt1/NLSR/KEY/ksk-5/ID-CERT/%01

http://named-data.net/doc/NLSR/current/SECURITY-CONFIG.html

Optimisation of Cache Distribution

- Default NDN caching strategy is to cache all packets, results in duplicated data around network
- More sophisticated approach will involve caching of the most popular data optimally across the network based on interest packets
- Less popular data likely not to be cached by routers

- E. Yeh et al. "VIP: A framework for joint dynamic forwarding and caching in named data networks," in Proc. 1st ACM Conf. Inf.-Centric Netw. (ICN), Paris, France, pp. 117–126, Sep. 2014 http://conferences2.sigcomm.org/acm-icn/2014/papers/p117.pdf