

# Supporting Climate Applications over Named Data Networking (NDN)

Christos Papadopoulos

David Randall

Cathie Olschanowsky

Kelley Wittmeyer

Susmit Shannigrahi

Don Dazlich

Steve DiBenedetto

Mark Branson

Network Partner: ESnet

Colorado State University

NDNcomm, Sept 5, 2014

#13410999: NSF CC-NIE Integration award

# Network Support for Climate Applications is Challenging

- Several hard problems: discovery, retrieval, replication, synchronization, versioning, security
- Various independent solutions, ESG/ESGF, Globus, NASA GCMD, BADC, NOMADS, USG Geo-Data Portal, many more
- No consistent naming structure
  - Major impediment in data management
  - .. but also major challenge in migrating to NDN

# Climate Naming Standardization Efforts

- The climate community has recognized the importance of structured naming
- One major example: Coupled Model Inter-comparison Project (CMIP)5

# CMIP5 Data Reference Syntax (DRS) and Controlled Vocabularies

- “.. a common naming system to be used in files, directories, metadata, and URLs to identify datasets wherever they might be located within the distributed CMIP5 archive. It defines controlled vocabularies for many of the components comprising the data reference syntax (DRS).”
- Current version is 1.3.1, June 2012
- CMIP6 efforts just starting

[http://cmip-pcmdi.llnl.gov/cmip5/docs/cmip5\\_data\\_reference\\_syntax.pdf](http://cmip-pcmdi.llnl.gov/cmip5/docs/cmip5_data_reference_syntax.pdf)

# DRS: Directory and File Structure

- Directory encoding:
  - /<activity>/<product>/<institute>/<model>/<experiment>/<frequency>/<modeling realm>/ <variable name>/<ensemble member>
  - Example:  
`/CMIP5/output/MOHC/HadCM3/decadal1990/day/atmos/tas/r3i2p1/`
- Filename encoding:
  - filename = <variable name>\_<MIP table>\_<model>\_<experiment>\_<ensemble member>[\_<temporal subset>][\_<geographical info>].nc
  - Example:  
`tas_Amon_HADCM3_historical_r1i1p1_185001-200512.nc`

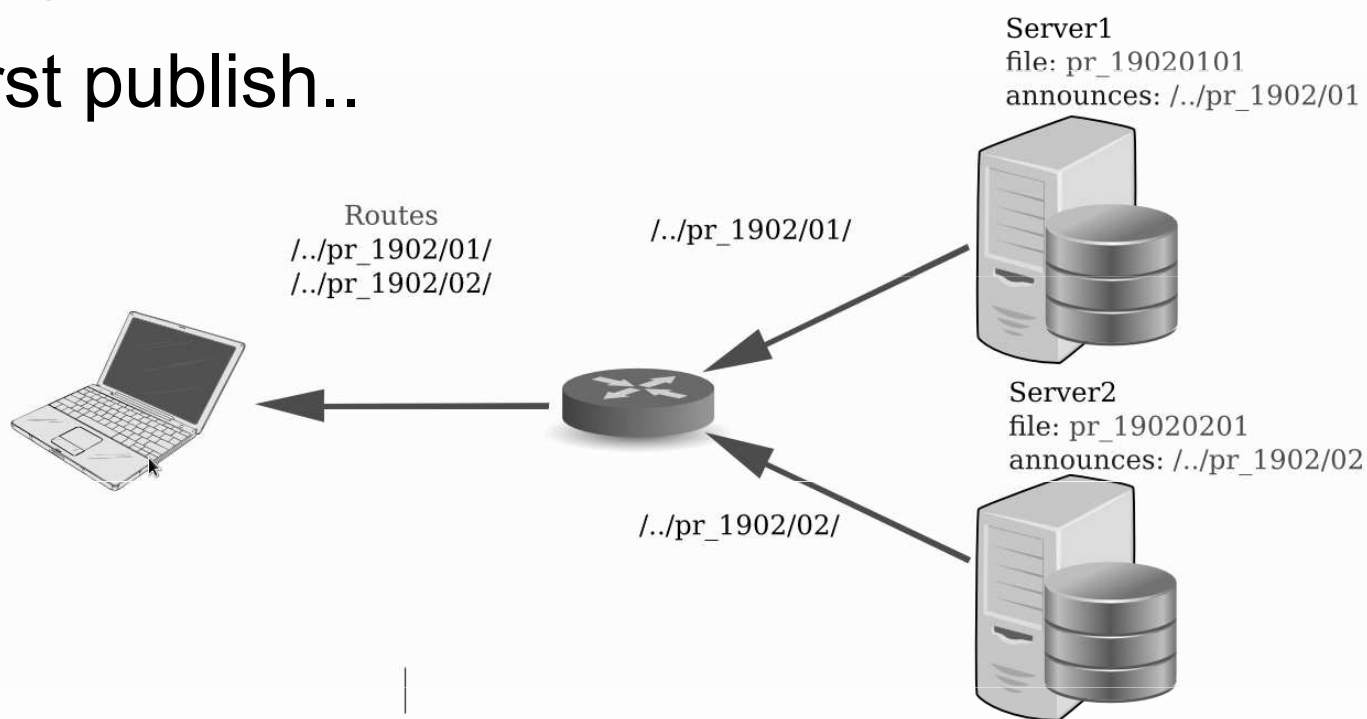
# Observations

- CMIP5 DRS is NDN-compliant
  - Hierarchical
  - Clearly defined name components
  - Well-defined vocabularies
  - Organizes components from less specific to more specific
  - Utilities to translate into DRS (CMOR)
  - Produces virtually ready-to-use NDN names
- ..but more of a stick than a carrot

# NDN in Climate Applications

Can we do more carrot and less stick?

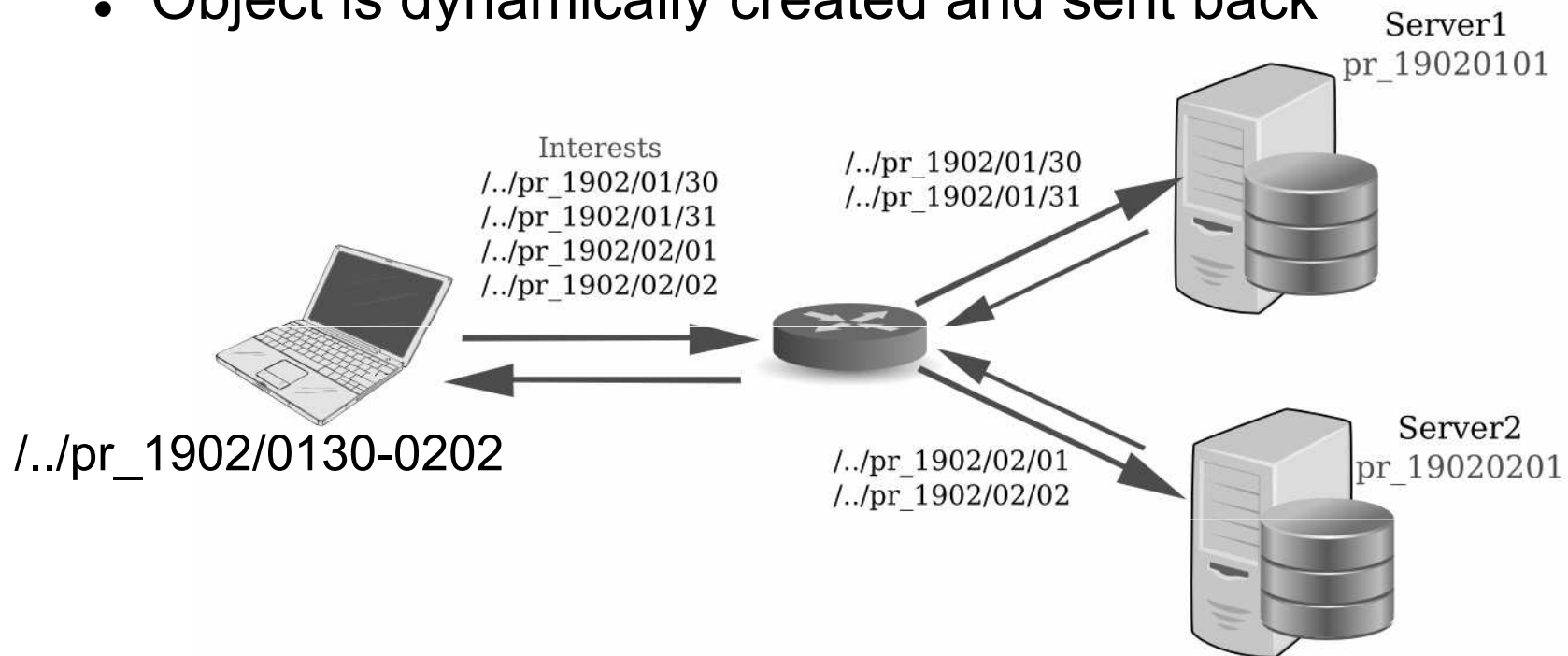
- NDN makes publishing, discovery and retrieval trivial..
- First publish..



# Name-Driven Computation

Then (more than just) discover and retrieve:

- Consumer wants subset for Jan30–Feb02
- Consumer composes object name and expresses interest, for example, `/../pr_1902/0130-0202`
- Object is dynamically created and sent back





# Reality Check

- In Real Life namespaces are much messier
  - Scientists use their own home-grown, ad-hoc naming schemes
- Even in CMIP5, information needed to translate names is often hard to determine
  - “..different calendars: Gregorian, idealistic 360-day, or assuming no leap years. “
  - “There are almost as many ways to store data as there are groups providing data.”

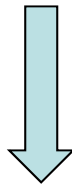
<http://www.realclimate.org/index.php/archives/2014/04/labels-for-climate-data/#sthash.V40L0Zg2.dpuf>

# Our Translator Architecture

h

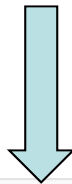
spcesm-ctrl.pop.h.1891-01.nc

Contents of file



translator

Filename to NDN  
name mapping  
schema and user  
defined components



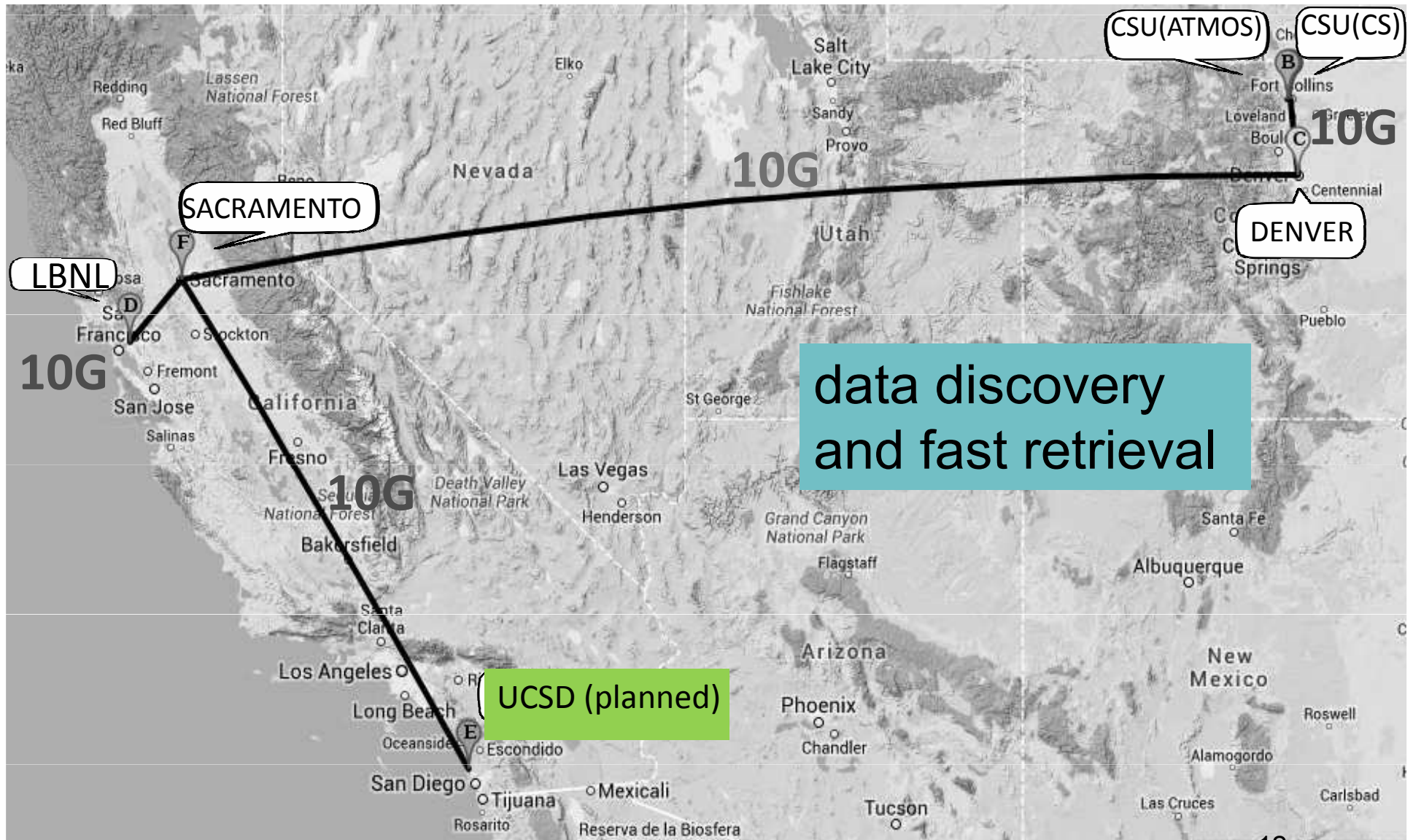
/coupled/control /CMMAPr3i1p1/spcesm-ctrl/pop/1M/1891-01/

activity/sub-activity/organization/ensemble/experiment/model/granularity/start-time

# Climate Testbed over ESnet

- Currently separate from the NDN testbed
- Three sites connected via NDN routers over ESnet
- ESnet is much more than just a hardware provider/host – also a research partner
  - Supports deployment, research, student interns and more
- State of the art Dell PE720xd servers with 10G fiber connections throughout
  - 20 cores, 128GB RAM, 48TB local storage

# Climate Testbed Locations



# Short Demos

- Publishing data
- Multiple publishers
- Failover

```
self.excludeList = Exclude()
removeComponents = 2
self.sendNextInterest(Name(initialInterest), None)
```

```
def onData(self, interest, data):
    self.excludeList.appendComponent(dataName[-(self.removeComponents)])
    self.dataList.append(dataName.getPrefix(dataName.size() - 1).toUri())
    self.sendNextInterest(dataName.getPrefix(dataName.size() -
self.removeComponents), self.excludeList)
```

```
def onTimeout(self, interest):
    self.removeComponents += 1;
    self.excludeList.clear()
    self.excludeList.appendComponent(interest.getName()[-1])
    interest_name = interest.getName().getPrefix(interest.getName().size() -
self.removeComponents).toUri()
    self.sendNextInterest(Name(interest_name), self.excludeList)
```







# Conclusions

- NDN offers improvements in two areas:
  - Reduce application development pain with better network services
    - A more sane way to interact with the network
  - Help standardize data naming by offering a new carrot: automatic content publishing
- Interactions with climate scientists have been very positive
- We see huge potential in many other application domains
- Stick around to watch NDN in a box..