NDN in Collaborative Big Data Applications

Christos Papadopoulos
Colorado State University
AsiaFI 2013
Hong Kong, Aug 11, 2013
Big Data Applications Requirements

• Very large datasets (TB – PB)
  – Simulations, measurements
  – Hard to keep track of different versions
  – Hard to move around (FEDEX rather than Internet)

• Locating Metadata
  – Extensive dataset descriptions
  – Notes, emails, etc.
  – Hard to locate
Current Network Solutions

- GridFTP for bulk transfer
- Various formats (e.g., netCDF) along with tool libraries
- Directory services/cataloging
- Various repositories with different rules, access controls
  - Need separate accounts for each repository
- Other complex solutions
Can NDN Help?

- **Question**: Can we use NDN to address many of these problems?
- **Approach**: Build a campus infrastructure (actually 3 campuses) and introduce NDN into the workflow of an existing big data application
- **Application**: Climate modeling by the CMMAP group at Colorado State
The Power of Naming

• Naming can fetch the data transparently regardless of location
  – Requests can go to appropriate place
• Naming can result in generation of new data
  – I can ask for data that does not yet exist!
  – ..as long as I know how to construct the name
Naming Example

• A project at CMMAP at CSU uses precipitation data from year 1902
• Create a global project name:
  – Colostate.edu/CMMAP/PrecipitationProj1
• Researchers agree on this global project name, which is added to the routing system
• Data under this project can be referred to as:
  – Colostate.edu/CMMAP/PrecipitationProj1/data/pr_1902/01
Naming Example (cont.)

• Suppose researchers add annotations to the data:
  – Colostate.edu/CMMAP/PrecipitationProj1/annotations/pr_1902/01/

• Or emails related to the project:
  – Colostate.edu/CMMAP/PrecipitationProj1/email/pr_1902/01/

• These are just simple examples – namespace design is a big issue in NDN
Experiment Topology
Experiment Setup

- Two servers and two clients
- Servers at CSU, clients at Memphis and St. Louis
- Nodes exchange routes using OSPFN
Announcements

- Servers have `.nc` files, each `.nc` file have one month’s data
- Route announcements in network are based on filename
- Each server advertises one prefix for a file
  - Server having file `pr_19020101.nc` announces `../../pr_1902/01/`
- OSPFN propagates announcement
Dynamic Data Generation

- Servers parse interest names and find the date range
- Pass date range to \textit{ncks} tool.
- \textit{ncks} tool extracts data, writes to file and returns the filename to server
- Server sends back file
An Example Data Request

- Want data for Jan 30 – Feb 02
- Client expresses interests, one for each day
- Interests for Jan 30-31 go to server1
- Interests for Feb 01-02 go to server2
- Data is dynamically generated and sent back
- Client consolidates reply and writes to disk
Repeat Requests and Cache

- If asked for same data, requests are answered from cache
- Saves transmission time, extraction time and transfer time
Partially Cached Data

- What happens if we ask for Jan 29 – Feb 2?
- Request for data not cached goes to server
- Rest is answered from cache
Collaborations

- A asked for data for Jan 30 - Feb 2.
- B later asks for same data.
- B receives data from cache.
Multipath Interest Forwarding

- Interests may be forwarded opportunistically to many destinations
  - Strategy Layer
- Data may be concurrently retrieved from multiple places
  - Bit torrent-style retrieval
NDN vs. Directory Services

• Why not just use IP, CDNs and directory services?

• You can, but..
  – Which directory service? Which directory to use? Which access protocol?
  – Where to place CDN caches?
  – How do you do backups?

• A very complex and hard to use solution
NDN vs. IP Data Retrieval

• Steps in IP
  – Locate the appropriate files (lookup in a catalog)
  – Convert requested data from human readable form into an index into the file
  – Extract the appropriate range from each file at each server
  – Fetch each extracted piece locally
  – Merge each piece into the requested dataset

• Steps in NDN
  – Ask for the requested object by name
Quality of Service

• NDN simplifies QoS implementation
• Symmetry between Interest/Data packets means reservations are easy to make
  – Many ideas in RSVP become easy to implement
• Approach:
  – Send Interest with name and reservation spec
  – Routers make temporary reservations
  – First Data packet confirms path reservations
  – Constant Interest/Data exchange allows for dynamic reservation updates
Delivery Estimation

• It is often desirable for researchers to know how long it would take to retrieve an object
  – The object is very big (simulation run)
  – The object has not been generated yet (e.g., a subset, a simulation that has not finished, etc.)

• Once researchers have delivery estimate, they can decide whether to request the object
Implementing Delivery Estimation

• Combined with QoS, NDN allows the network to answer questions such as:
  – How will it take to retrieve object N?

• Approach:
  – Send Interest asking for availability of object N not retrieval)
  – Interests also carry an estimate of available bandwidth on return path
  – Network responds with cached (portions of) object N
  – Endpoints respond with either availability or time to generate object N

• Possible because NDN uses names – it would be too complex otherwise
Integrating with Existing Workflows

- Challenge: how to integrate NDN with current researcher workflow?
- Cannot be (too) disruptive to the current workflow
- Researchers are used to working with files
- Can we build a veneer over NDN?
Step 1: Integrate with the Filesystem

• Create an NDN-aware filesystem
  – Build using a user-space filesystem
• Add new, relevant File attributes
  – Shared, versioned, backed up, synchronized, etc.
Step 2: Build Translators

• Problem: Going from any file name to NDN-appropriate name

• One solution: Translators
  – Handle various home-grown namespaces, often specific to researcher or research group to NDN names
  – Migrating to potentially richer semantics
  – Eases transition to NDN – just run the translator on your existing files
  – Adds structure to what is currently ad-hoc naming practices – but will it gain acceptance?
Conclusions

• We believe that NDN is a good fit for many applications
• We are currently trying one application domain – big data
  – Huge datasets – network intensive
  – Collaborative applications
  – Established, though largely ad-hoc workflows
  – Needs powerful search/location services
  – Relatively low number of users (compared to Internet)
  – Open to new solutions