Impact of Content-Centric Networking on Large-Scale Scientific Applications

Hiroyuki Ohsaki

Department of Information Networking
Graduate School of Information Science and Technology
Osaka University, Japan

September 18, 2012

Data-Centric Networking

Data-centric networking is an emerging communication paradigm, which is expected to overcome several limitations of the conventional IP network

Host-centric networking (conventional)

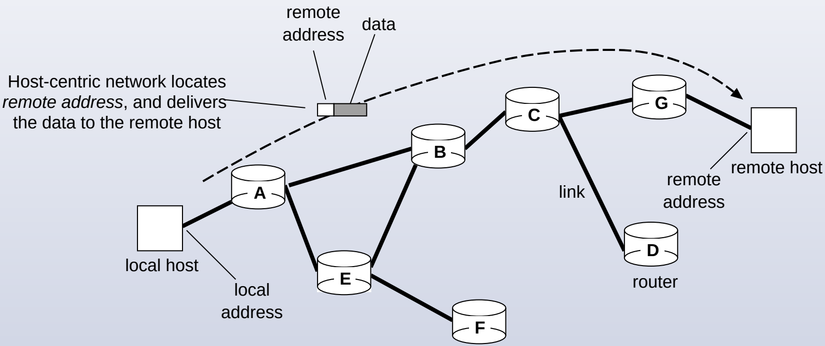
- End system to be connected via the network is the first-class object
- A stream of data is *delivered to the host*, which is uniquely identified by its identifier (e.g., IP address), via the network
- Example architectures: IP, Ethernet, MPLS, ATM, FDDI

Data-centric networking

- Data transferred in the network is the first-class object
- *Requested content*, which is uniquely identified by its identifier (e.g., content name), are retrieved from the network
- Example architectures: DONA, CCN, NDN

Host-Centric Networking

End system to be connected via the network is the first-class object

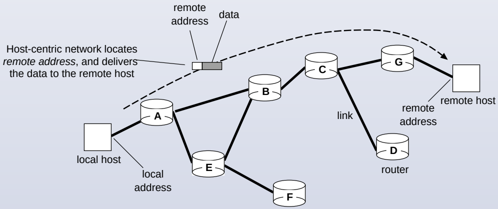


User's View on Host-Centric Networking (1/2)

Host-centric networking provides a *virtual link* between end hosts, each of which is identified with a unique identifier *address*

Data transfer between end hosts is simple:

1. A user sends a data destined for *remote address*
2. A host-centric network locates *remote host* corresponding to *remote address*, and delivers the data to the remote host



User's View on Host-Centric Networking (2/2)

However, content retrieval is more complicated...

1. A user requests a content named `http://wwww.ispl.jp/photo.jpg`
2. Local host resolves *remote address* corresponding to `www.ispl.jp` with DNS (Domain Name System)
3. Local host establishes a TCP connection between *remote address* and *local address*
4. Local host sends a request for `/photo.jpg` to *remote address*
5. A data-centric network locates *remote address*, and delivers the request to the remote host
6. Remote host sends the JPEG file corresponding to `/photo.jpg` back to *local address*
7. A data-centric network locates *local address*, and delivers the content to the local host

Issues in Host-Centric Networking

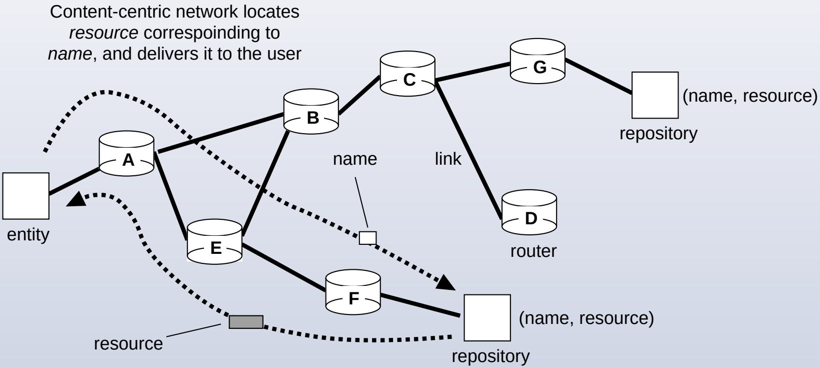
Host-centric networking is good for providing *communication channels* between end hosts (e.g., telnet, VoIP)

But, host-centric networking is not good for *data-intensive applications* (e.g., Web, database, video/audio streaming, network storage) because users are interested not in the location of the content but in the content itself

- Inefficiency
  - Non-optimized content delivery
  - Non-reusable content
- Low availability
  - Content server is the single point-of-failure
- Insufficient security
  - Content is not authenticated
  - No standard security mechanism in IP

Data-Centric Networking

Data transferred in the network is the first-class object



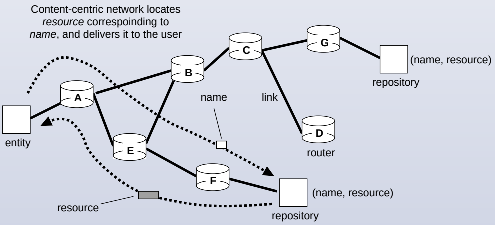
User's View on Data-Centric Networking (1/2)

Data-centric networking is essentially a network-level realization of (*key*, *value*)-pair database (c.f., NoSQL, MapReduce)

- Every content is represented as a (*name*, *resource*)-pair

Data transfer between end hosts is not supported
Content retrieval is very simple:

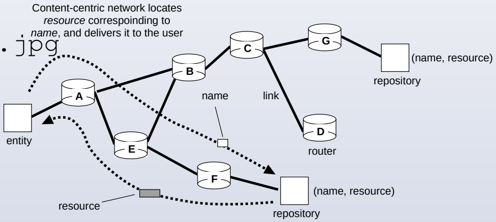
1. A user requests a content named *name*
2. A data-centric network locates *resource* corresponding to *name*, and delivers it to the user



User's View on Data-Centric Networking (2/2)

Content retrieval example:

1. A user requests a content named `ccnx://ccn.org/photo.jpg`
2. A data-centric network locates any (usually, nearest) copy of the JPEG file named as `/ccn.org/photo.jpg`, and delivers the file to the user



Major Data-Centric Networking Architectures

Data-centric networking has been recently studied in the literature, and there are several data-centric network architecture proposals

- DONA (Data-Oriented Network Architecture) [Koponen07:SIGCOMM]
  - Network topology: tree (DNS-like content lookup)
  - Content name: GUID (Globally Unique Identifier)
- CCN (Content-Centric networking) [Jacobson09:CoNEXT]
  - Network topology: arbitrary (IP-like content routing)
  - Content name: URI (Uniform Resource Identifier)
- NDN (Named-Data Networking) [Zhang10:NDN_Project]
  - Essentially same with CCN

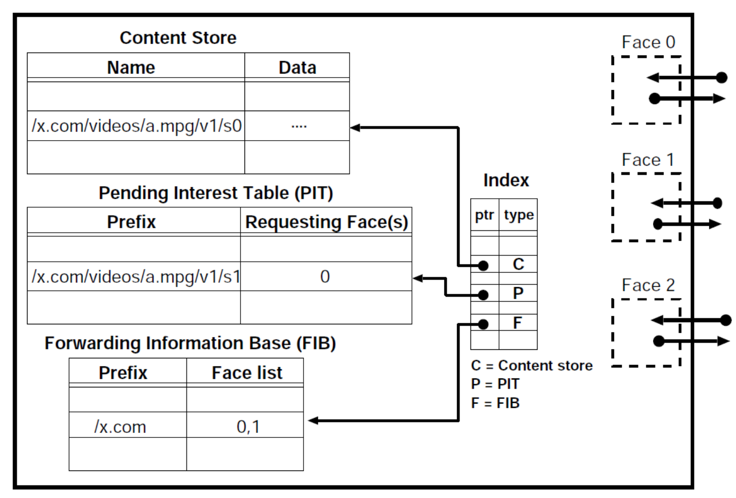
CCN (Content-Centric Networking) Overview

CCN (Content-Centric Networking) is one of major data-centric networking architectures developed by Xerox PARC, and it adopts a request-and-reply communication model

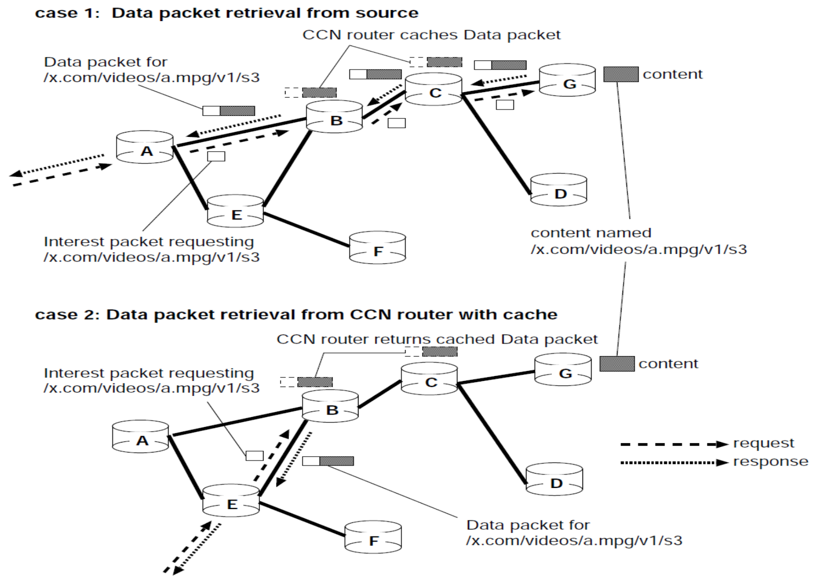
Content retrieval example:

1. A user injects an Interest packet for the content named `/ccn.org/photo.jpg`
2. A CCN router (selectively) floods the Interest packet to neighbor CCN routers according to their content routing tables (i.e., FIB (Forward Information Base))
3. If the JPEG file named as `/ccn.org/photo.jpg` is found at any CCN router, the file is delivered to the user as a Data packet by reversely traversing the path
4. The file is cached in the CCN router's buffer cache (i.e., ContentStore) for later reuse

CCN Router Structure [Jacobson09:CoNEXT]



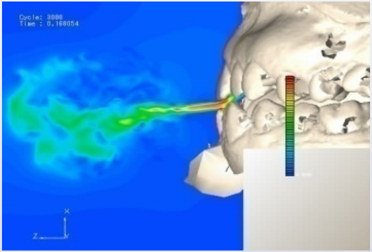
Routing of Interest and Data packets in CCN

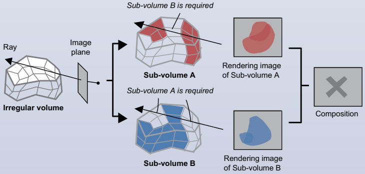


Large-Scale Scientific Applications

Scientific applications are generally large-scale and data-intensive

- Parallel simulation
  - Numerical, event-based, and agent-based simulations of a large and complex system
- Large-scale remote visualization
  - Volume-based and particle-based renderings of a large and complex object
- Big-data analysis
  - Distributed large-volume data analysis and processing





Issues in Large-Scale Scientific Applications (1/2)

Scientific applications are generally large-scale and data-intensive, which makes it still challenge to run those applications in distributed and massively parallel way

- Complexity
  - A user has to manage both networking and computing resources by himself/herself
  - Several middlewares (e.g., Grid and Cloud middlewares) partly help management issues, but those middlewares are complex and not transparent to users
- Inefficiency
  - Manual resource management are difficult to optimize; resource usage are usually not efficient
  - Middlewares are built in an upper-layer and they are generally not aware of network status, resulting in inefficient network resource usage

Issues in Large-Scale Scientific Applications (2/2)

- Low scalability
  - Neither manual resource management nor middleware is scalable due to their complexities
- Low availability
  - Scientific applications are prone to network failures since they are independent of network status

Research Questions

What are the impacts and implications of data-centric networking on large-scale and scientific applications?

- How can complexity, efficiency, scalability, availability of scientific applications be improved with data-centric networking? Why? (performance studies)
- How should scientific applications be designed and implemented in order to take advantage of data-centric networking? (application design methodology)
- What type of mechanisms/features are necessary in data-centric networking to optimize the performance/scalability/availability/usability of scientific applications? (network architecture study)

Impact of CCN on Large-Scale Scientific Applications

Scientific applications must be significantly benefitted from data-centric networking, but its impact and implications have not been clarified or understood

For instance, data-centric networking could…

- Simplify the *design and implementation* of scientific applications
- Improve both *computing and networking efficiency* of scientific applications
- Improve *scalability* of scientific applications because of simplicity and efficiency
- Improve *availability* of scientific applications under non-negligible failures

CCN Advantages to Large-Scale Scientific Applications

- High Availability
  - *CDN (Content-Delivery Network)-like operation* realizes high content availability because of arbitrary number of content replicas (sources) and content caching
- High Efficiency
  - *In-network content caching* at nodes (e.g., routers) minimizes content delivery delay and also reduces the amount of network traffic
- Usability
  - *Intuitive UNI (User-to-Network Interface)* for data-intensive applications (e.g., Web, database, audio/video streaming, network storage)

CCN Disadvantages to Large-Scale Scientific Applications

- Backward incompatibility
  - Applications must be redesigned to take the advantages of CCN
- Immaturity
  - Lack of standard APIs
  - Lack of infrastructure
    - There are only application-level experimental implementations of CCN
    - Many open and unresolved issues to deploy
  - Lack of applications

CCN Content Availability Analysis

Notations

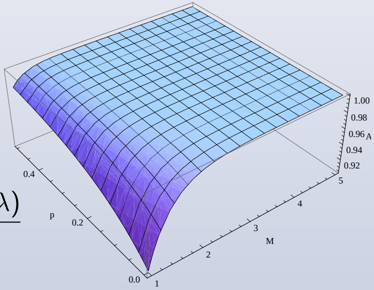
- M: the number of content replicas
- H: the average number of hops between user and repository
- λ: node failure rate
- p: CCN router cache hit rate

Result

- The content availability A

$$A = 1 - (1 - a)^M$$

$$a = \frac{\lambda (1 - \lambda)^H (1 - p)^H + p (1 - \lambda)}{\lambda + p (1 - \lambda)}$$



$\lambda = 0.01, H = 10$

CCN Content Delivery Efficiency Analysis

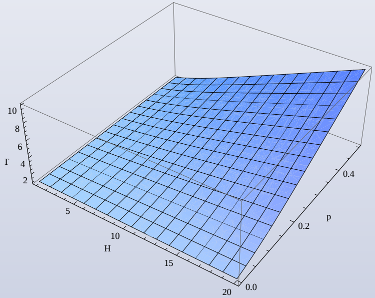
Notations

- H: the average number of hops between user and repository
- p: CCN router cache hit rate
- $\widetilde{T}$: throughput without content caching

Result

- Throughput T

$$T = \frac{p\, H\, \widetilde{T}}{1-(1-p)^H}$$



$\widetilde{T} = 1.0$

Example Application 1: CCN-based remote procedure call

Data-centric networking can be seen as an ultra light-weight and near optimal remote procedure call

- Data-centric networking is essentially a distributed (*key*, *value*)-pair database
- So, remote procedure call in scientific applications (e.g., Sun RPC, IPC, CORBA, MPI) can be realized simply with CCN
  - *key*: combination of the procedure name and arguments to the procesure (e.g., `Inverse[{{1,2},{3,4}}]`)
  - *value*: output from the procedure (e.g., `{{-2,1},{1.5,-0.5}}`)
- Merits: simplicity, efficiency, scalability, high availability

Example Application 2: CCN-based distributed filesystem

Very efficient and highly reliable distributed file system can be realized with data-centric networking

- Again, data-centric networking is essentially a distributed (*key*, *value*)-pair database
- So, distributed filesystem needed for scientific applications (e.g., NFS, CIFS, WebDAV, Gfarm) could be realized on top of CCN
  - *key*: the path name of a file
  - *value*: the file content or the directory entry
  - But cache consistency management is necessary
- Merits: efficiency, scalability, high availability

Conclusion

Data-centric networking is an emerging communication paradigm, which is expected to provide efficiency, high availability, and security.

- Issues in large-scale scientific applications (i.e., complexity, inefficiency, low scalability, low availability) could be solved with data-centric networking
- Major research areas are addressed
  - Performance studies
    - CCN content availability analysis
    - CCN content delivery efficiency analysis
  - Application design methodology
    - CCN-based remote procedure call
    - CCN-based distributed filesystem
  - Network architecture study