OpenFlow-based Content-Centric Networking Architecture and Router Implementation

Atsushi Ooka
Osaka University
Japan

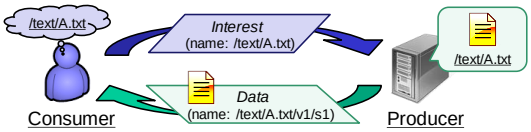
Outline

- Research area
  - Content-Centric Networking (CCN)
- Objectives
  - Resolve Implementation and Deployment Issues
- Research Approach
  - OpenFlow-based CCN
- Major Outcomes
- Conclusion

Content-Centric Networking (CCN)

- Content-based addressing with "*name*"
- Transmission using *Interest* and *Data* packets
- Natively supported mechanisms to disseminate contents
  - Interest aggregation, data multicasting, In-network cashing

➡ Realization of CCN is a big challenge

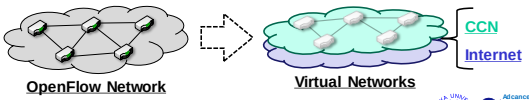


Objective and Approach

- Objective: Realization of CCN
  - Implementation of CCN forwarding
  - Deployment of CCN

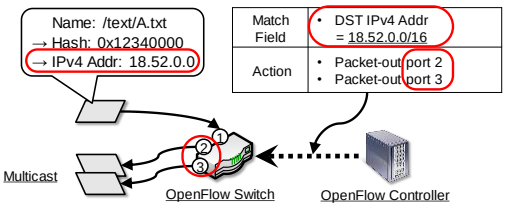
⬇

- Approach: Implement OpenFlow-based CCN
  - Allows to easily implement new protocols
  - Allows multiple systems to share one network



Design of OpenFlow-based CCN

- Fast Processing based on name address
  - Convert name to hash value (that is hierarchically structured)
  - Write the hash in IPv4 address field
- Multicast using multiple actions



Major Outcomes

- Hierarchically structured hash value
  - Support active name
- Analysis of hash collision probability
  - Explore the trade-off
- Implementation of the program for OpenFlow controller
  - Show the demonstration

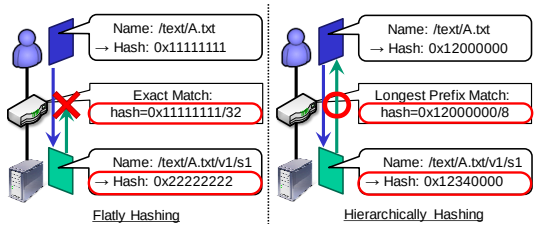
Hierarchically Hashed Name

- Assign 4 bits to each component
  - Ex. /text/A.txt/v1/s1
  - "text" = 2, "A.txt" = 1, "v1" = 4, "s1" = 2
  - IP address: 33.66.0.0/16 (= 0x21420000)
- Trade-off between *B* and *C*
  - *B*: number of bits assigned to component
  - *C*: number of components that can be hashed

Protocol	*C* = number of components			
	B = 4	*B* = 8	*B* = 16	*B* = 32
IPv4 (32 bit)	8	4	2	1
MAC (48 bit)	12	6	3	2
IPv6 (128 bit)	32	16	8	4

Support Active Name

- Resolve a mismatch caused by flatly hashing[1]



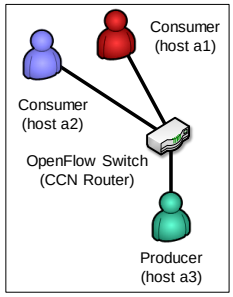
[1] N. Blefari-Melazzi, A. Detti, G. Mazza, G. Morabito, S. Salsano, L. Veltri, "*An OpenFlow-based Testbed for Information Centric Networking*," in Proceedings of Future Network & Mobile Summit 2012, 4 – 6 July 2012, Berlin, Germany

Hash Collision Probability

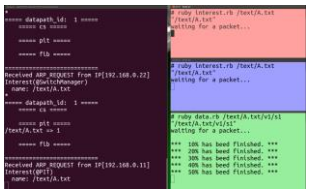
- $P_4^{\text{Protocol}} = \left(1 - \prod_{i=1}^{N} \frac{(2^B - 1) - i}{2^B - 1}\right)^{\frac{len}{B}}$
  - *B*: number of bits assigned to component
  - *N*: number of names
  - *len*: bit-length of the address in the protocol

Protocol	N(p = 0.1) / N(p = 0.9)		
	B = 4	*B* = 8	*B* = 16
IPv4 (32 bit)	6 / 11	21 / 43	224 / 624
MAC (48 bit)	7 / 11	27 / 47	287 / 664
IPv6 (128 bit)	9 / 12	32 / 50	427 / 753

Program Demonstration

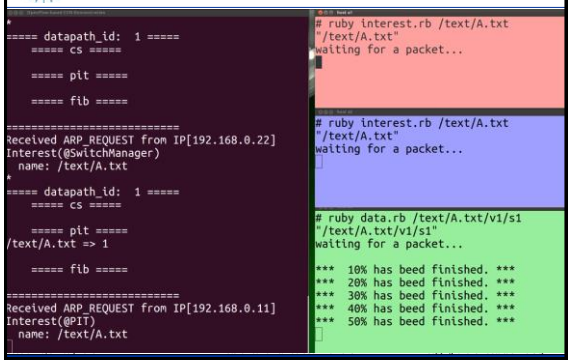


Demo Environment



Screenshot of Demo

Program Demonstration



Conclusion and Future Work

- Conclusion
  - Feasibility of OpenFlow-based CCN
    - Provides the implementation of CCN forwarding on OpenFlow
  - Support active name
  - Analyze hash collision probability
- Future work
  - Implement lacking mechanisms
    - Routing, hardware cashing
  - Explore scalability
    - Hash collision probability, flow entries
  - Evaluate performance using hardware

