I-CAN Workshop on Information-Centric Access Networks, Athens, June 2-5, 2015

Service Centric Networking

Torsten Braun
Communication and Distributed Systems
Institute of Computer Science and Applied Mathematics
University of Bern, Switzerland
braun@iam.unibe.ch, cds.unibe.ch
Traditional Web Retrieval / Web Services

- User's end system
- DNS server
- Search engine / service registry
- Web server / web service
Motivation

> Information-Centric Networking focuses on efficient transfer of content, supported by caching.

> The Internet evolves to an infrastructure supporting more general services, e.g., web services, IoT, cloud, etc.

> Several proposals to support service discovery and efficient data transfer between service consumer and service provider.
  — not using ICN: Serval
  — using ICN:
    SCN/NextServ, NFN, CCNxServ, SOCCER, SOFIA, XIA, FCSC

> Specific application scenarios
  — Data Centre Networking
  — Do-It-Yourself (DIY) networks
  — Internet of Things (IoT)
NSDI 2012

Serval: An End-host Stack for Service-Centric Networking

Erik Nordström, David Shue, Prem Gopalan, Robert Kiefer, Matvey Arye, Steven Ko, Jennifer Rexford, and Michael Freedman
Serval

- Service Access Layer (SAL) between the transport and network layers.
- SAL maps service names in packets to network addresses, based on rules in its service table.
- SAL’s position below transport layer provides a programmable service-level data plane that can adopt diverse service discovery techniques.
- SAL performs signalling between end-points to establish / migrate flows.

![Serval Diagram]

[Legend: TCP/IP - bind (IP + port), demux (IP + port); Serval - bind (serviceID), resolve (serviceID), demux (flowID), forward (IP)]
Serval Packet Format

Torsten Braun: Service-Centric Networking

I-CAN, Athens, June 4, 2015
Serval Node Architecture

- Application
  - Socket
  - bind()
  - close()
- User Space
- Service Controller
- Remote Service Controller
- Active Sockets
- Kernel
  - Network Stack
- FlowID
- Socket
- ServiceID
- Action
- Sock/Addr
- Flow Table
- Service Table
- IP Forwarding Table
- Dest Address
- Next Hop
- SYN Datagram
- Transport
- Service Access
- Network
- Data Delivery
Serval Forwarding

- **FORWARD rules**
  - indicate a set of one or more IP addresses (service routing)

- **DEMUX rules**
  - are used by recipients to deliver a received packet to a local socket, when an application is listening on the serviceID

- **DELAY rules**
  - cause the stack to queue the packet and notify the service controller of serviceID, e.g., allowing a rule to be installed “on-demand”.

- **DROP rules**
  - simply discard unwanted packets.
Serval Connection Establishment
ICC Workshop on Future Networks, 2011

Service-Centric Networking

Torsten Braun, Volker Hilt, Markus Hofmann, Ivica Rimac, Moritz Steiner, and Matteo Varvello
Service-Centric Networking (SCN)

> CCN is content-centric and encodes a few operations on content as extensions of names.
  — %C1.org.ccnx.frobnicate~1~37: command in namespace org.ccnx, with operation frobnicate, which takes two arguments: 1, 37

> Proposal: Service-Centric Networking
  — Extension of content-centric networking to support services, possibly operating on content.
  — Description of a service using content naming scheme, e.g., /google.com/file-service
  — Services are provided by active components: service elements.
  — Service request/response in Interest/Data message
Target Services Supported by SCN

- Client-oriented services, e.g., web services
- Continuous content retrieval and streaming services, e.g., A/V conferencing, streaming
- Event services, e.g., exceeding sensor or stocks data thresholds
- In-network services, e.g., aggregation / filtering of (sensor) data
- Infrastructure services, e.g., cloud computing services
  - find/call cloud services (computation, storage)
  - deploy cloud services
  - support caching of data stored in the cloud
Service-Centric Network

Service request/response

SCN router
Advantages of SCN

- No service lookup and service registry
- Caching of service data
- Extended caching of multimedia data (transcoding)
- Location-based services
- Optimized service selection
- Lower delay by in-network services
Uniform Naming for Services (Functions) and Content (Data)

> Services perform (data) processing and are represented by functions to be invoked. Content stores for data.

> Service-centric networking should support both data and functions.

> Object-orientated programming paradigm integrates both functions and data into objects. Method calls among objects to invoke functions.

> Proposal: Object names for both services (functions) and content (data), e.g.,
  — /youtube.com/rendering
  — /unibe.ch/braun/talks/i-can

> Advantages of object-oriented approach
  — Uniform naming
  — Services can be implemented as a set of cooperating objects
SCN Object Types

1: Content Object
   - read
   - content data

2: Service Object
   - function1
   - function2
   - function3

3: Content/Service Object
   - read
   - content data
   - function1
   - function2
Example: Video Rendering

client

render

read

write

video-file1

read

write

render

video-file2
Example: Real-time Audio Conferencing

- Echo cancellation
- Trans-coding
- Mixing

Sent audio data → Echo cancellation → Trans-coding → Mixing

Received audio data
Real-Time Audio Conferencing Service

I-CAN, Athens, June 4, 2015
NextServe Framework: Supporting Services over Content-Centric Networking

Dima Mansour, Torsten Braun, and Carlos Anastasiades
NextServ Naming and Operation

$fileManager/zip($university/profile.pdf)$

$fileManager/zip/($university/profile.pdf)$

$scn/encrypt(“P@ssw0rd”, fileManager/zip(university/profile.pdf))$
ACM SAC 2013

Service-Centric Networking Extensions

Torsten Braun, Andrea Mauthe, and Vasilios Siris
SCN Prototype Evaluation

- Image conversion service running on an unmodified CCNx router
- CCNx router can transcode images stored in cache or repository.

1-8: Client 1 retrieves bit map image (36 MB)
9-16: Client 2 retrieves JPEG image (6 MB)
17-20: Client 1 retrieves JPEG image
SCN Support by Different ICN Approaches

Name Resolution and Data Transport in ICN

> Decoupled, e.g., PSI
  — Name resolution and data transport are independent of each other with possibly different nodes for resolution and data transport.
  — allows some more flexibility for service name resolution
  — supports considering parameters and parameter types for service selection

> Tightly Coupled, e.g., CCN
  — Nodes for both name resolution and data transport with inverse data path compared to search path
  — easy service routing support by extension of routing tables (Forwarding Information Bases, FIBs) by routing metric, cf. SoCCeR
An Information Centric Network for Computing the Distribution of Computations

Manolis Sifalakis, Basil Kohler, Christopher Scherb, and Christian Tschudin
Named Function Networking (NFN)

- Names in NFN represent functional programs in their simplest, most compact form: λ-calculus expressions.
- Lambda calculus (basis of functional programming) defines recursively the form of terms that compose a valid expression in one of three cases:
  - variable lookup
  - function application
  - function abstraction

expr ::= v | expr-l expr-r | λx.expr
  - v: variable
  - a simple function call expr-l(expr-r) with one argument
  - λx.expr
    - definition of a function with one argument.
    - consists of a λ-expression expr, in which all occurrences of the parameter x are the places, where the actual parameter value (function argument) has to be substituted.

- Example: application of a transcoding function on the media content:
  - λy.(/name/of/transcoder y) /name/of/media

- NFN extends the CCN architecture by
  - integrating a λ-expression resolution engine in a CCN relay
  - optionally hosting an application processing/execution environment.
IEEE ICC 2012

CCNxServ: Dynamic Service Scalability in Information-Centric Networks

Suman Srinivasan, Amandeep Singh, Dhruva Batni, Jae Woo Lee, Henning Schulzrinne, Volker Hilt, Gerald Kunzmann
CCNxServ

> naming scheme: contentname+servicename, e.g., ccnx://video.mp4+ad
> Content router is able to parse the name. Corresponding service is invoked on content name.
> Possibly content and service module have to be retrieved via CCNx before service invocation.
> Processed content is sent back to the client and places the processed content back into the CCNx namespace so that future requests for this combination of contentname+servicename will directly be able to fetch the processed content.
> based on NetServ service virtualization framework (using Click routers)
SoCCEr: Services over Content-Centric Routing

Shashank Shanbhag, Nico Schwan, Ivica Rimac, and Matteo Varvello
Motivation

- When multiple copies of a content item are available in the network, a request can be forwarded to all the copies resulting in redundant retrievals. This is too expensive if service requests result in redundant consumption of service node resources.
- CCN needs to be extended with mechanisms that can prevent duplicate invocation of services for one request.

Overview

- SoCCeR works as a control layer on top of CCN and manipulates the underlying Forwarding Information Base (FIB) to support “service selection,” i.e., routing a service request to the best service instance at any point in time.
- FIB entry for a service always points to a preferred instance of the service according to some metric (service load, path congestion, etc).
- SoCCeR is based on Ant Colony Optimization.
SoCCeR: Node Design

### SoCCeR

**Pheromone table**

<table>
<thead>
<tr>
<th>Service S</th>
<th>face</th>
<th>pheromone</th>
<th>probability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>$\rho_{1S}$</td>
<td>$P_{0S}$</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>$\rho_{2S}$</td>
<td>$P_{1S}$</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>$\rho_{3S}$</td>
<td>$P_{2S}$</td>
</tr>
</tbody>
</table>

**Service T**

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>$\rho_{0T}$</th>
<th>$P_{0T}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>$\rho_{2T}$</td>
<td>$P_{2T}$</td>
</tr>
</tbody>
</table>

### CCN (fast-path)

**FIB**

<table>
<thead>
<tr>
<th>name</th>
<th>face</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>0,2,4,7</td>
</tr>
<tr>
<td>S</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>2</td>
</tr>
<tr>
<td>C2</td>
<td>1,4</td>
</tr>
</tbody>
</table>
SoCCeR Ants

> Probabilistic Interest ant forwarding
> Response by Data ant includes service status information (service load, memory, etc.).
> Delay calculation and pheromone updates
> Evaporation of pheromones
SOFIA: Toward Service-Oriented Information Centric Networking

Qinghua Wu, Zhenyu Li, Jianer Zhou, Heng Jiang, Zhiyang Hu, Yunjie Liu, and Gaogang Xie
Applications establish service sessions consisting of multiple service connections, which are bound to specific servers.

Initiating service sessions by forwarding service requests

Service relays
- Redirection of service requests to specifically installed services dependent on configured rules
- Caching
- Translation between transport policies, e.g., in wireless/wired networks

Service announcements by servers to routers
SOFIA: Establishing Service Sessions

Packet header

Network layer

Src addr  Dest addr  Serv name  Local serv inst  Remote serv inst

Service layer

(c, a, X, XI, I)

(a, NIL, X, I, NIL)

Routing on service name

Routing on network address

I-CAN, Athens, June 4, 2015
XIA: Architecting a More Trustworthy and Evolvable Internet

David Naylor, Matthew Mukerjee, Patrick Agyapong, Robert Grandl, Ruogu Kang, Michel Machado, Stephanie Brown, Cody Doucette, Hsu-Chun Hsiao, Dongsu Han, Tiffany Hyun-Jin Kim, Hyeontaek Lim, Carol Ovon, Dong Zhou, Soo Bum Lee, Yue-Hsun Lin, Colleen Stuart, Daniel Barrett, Aditya Akella, David Andersen, John Byers, Laura Dabbish, Michael Kaminsky, Sara Kiesler, Jon Peha, Adrian Perrig, Srinivasan Seshan, Marvin Sirbu, and Peter Steenkiste
XIA Addressing

> Based on Internet principles, e.g., packets, address-based forwarding, narrow-waist at internetwork layer

> Addresses based on self-certifying XIA identifiers (XIDs)
  — Host XIDs: hash of public key
  — Service XIDs define what entities do.
  — Content XIDs: hash of content
  — Network XIDs: describe a network / domain

> Address DAGs
  (directed acyclic graphs)
XIA Mobility Support
ACM ICN 2014

Exploiting ICN for Flexible Management of Software-Defined Networks

Mayutan Arumaithurai, Jiachen Chen, Edo Monticelli, Xiaoming Fu, and K. K. Ramakrishnan
Function-Centric Service Chaining (FCSC)

> Supported example functions
  — Deep packet inspection, e.g. /DPI
  — Network address translation, e.g., /NAT
  — Proxying
  — Transparent caching
  — Accounting
  — Example: chain:/DPI/NAT/R5

> Proposal
  — Adding a naming layer to Software Defined Networking (SDN)
  — Adding flow states to packet header instead of switches
  — Packets include a list of functions in form of an ICN name
  — Function name is removed after applying a function
  — New functions can be added.
FCSC Operation
ICN 2012

An Information-Centric Architecture for Data Center Networks

Bong Jun Ko, Vasileios Pappas, Ramya Raghavendra, Yang Song, Raheleh Dilmaghani, Kang-Won Lee, Dinesh Verma
Centralized IC-DCN Architecture
IC-DCN Controller

> Routing Service
  — collects network topology information and content locations.
  — decides about Interest routing paths (label-based forwarding, source routing) and configures switches.
  — Data are routed on reverse path.
  — Path establishment tries to exploit caching.

> Naming Service
  — Assignment and management of name spaces to data producers

> Policy Service
  — Management policies
2015 Workshop on Do-It-Yourself Networking

SCANDEX: Service Centric Networking for Challenged Decentralised Networks

Arjuna Sathiaseelan, Liang Wang, Andrius Aucinas, Gareth Tyson, and Jon Crowcroft
SCANDEX (SCNx)

- Approach for Do-It-Yourself (DIY) networks
- Deployment of services in challenged environments and intermittent networks, e.g., DTNs
- Services can be set up on demand.
- Implementation of services using light-weight virtualization environments such as unikernels and Linux containers.
SCANDEX Network

- **Service Execution Gateways (SEG)**
  - attachment points for clients and servers, e.g., in wireless access points or base stations.
  - hosting and execution of services on behalf of its attached clients.

- **Forwarding Nodes (FN)**
  - responsible for routing requests for services towards available copies.
  - FNs cache services locally, but do not execute services.

- **Edge Gateways (GW)**
  - responsible for connecting different domains (two separate networks).
  - can temporarily function as Publishers and Subscribers of services, i.e., request services from another network on behalf of one of its users.

- **Brokers** perform service resolution and are responsible for intra-domain forwarding.
SCANDEX Operation

Example: A client node in island A wishes to access a service located in island B.

> A service request is passed from client to SEG.
> If the SEG possesses a copy of the service locally in its cache, the request is executed immediately.
> If the SEG does not have a local copy, the request must be forwarded via FNs to a Broker.
> Each DIY network must contain at least 1 Broker, which is responsible for indexing all services that are available (intra-domain).
> If the Broker knows an intra-domain instance of the service, it contacts the device that has the instance and requests service migration using a specific transmission module that is suitable for the underlying network topology, e.g., IP, Bundle protocol.
> If the Broker does not know of any copies of the service within its own network, it is necessary to forward the Interest on an inter-domain level via the GW, e.g., via Island B’s GW.
> Finally, the SEG instantiates the migrated instance and serves it to its clients.
A Solution for the Naming Problem for Name-Centric Services

Torsten Teubler, Mohamed Hail, and Horst Hellbrück
CCNx for WSNs

> Development of lightweight CCN implementation for wireless sensor networks

> Problems to be solved:
  — appropriate naming scheme and name configuration
  — Avoidance of verbose names in resource-constrained devices

> Example:

/fh-luebeck.de/building-18/floor-2/room-14/climate/%C1.de.manufacturer.sensors.humidity

<table>
<thead>
<tr>
<th>Domain</th>
<th>Site</th>
<th>Service Category</th>
<th>Manufacturer-Provided Command Name</th>
</tr>
</thead>
</table>

> Example: link-local service names

/%03%FB/climate/%C1.de.manufacturer.sensors.humidity

<table>
<thead>
<tr>
<th>Node ID</th>
<th>Service Category</th>
<th>Manufacturer-Provided Command Name</th>
</tr>
</thead>
</table>
ICN WSN Architecture
ACM ICN 2014

Information Centric Networking in the IoT: Experiments with NDN in the Wild

Emmanuel Baccelli, Christian Mehlis, Oliver Hahm, Thomas Schmidt, Matthias Wählish
ICN in IoT

> Implementation of CCN-Lite in RIOT, an operating system for constrained devices, without putting CCN on top of IP.

> Design and implementation of routing schemes for IoT
  — Interest Flooding
  — Reactive Optimistic Name-based Routing (RONR)
    – Initial interest flooding and auto-configuration of FIB entries on reverse path of data transmission
Conclusions
Conclusions

> Proposals for Service-Centric Networking (not) using ICN, but ICN seems to be quite beneficial.

> Several relevant application areas exist, e.g., IoT, DCN.

> Several issues in Service-Centric Networking require more research, e.g.,
  — Service management
  — Naming
  — Routing
Thanks for your attention!

> braun@iam.unibe.ch
> cds.unibe.ch