Service-Centric Networking for the Developing World

Arjuna Sathiaseelan, **Liang Wang**, Andrius Aucinas, Gareth Tyson*, Jon Crowcroft

N4D Lab liang.wang@cl.cam.ac.uk

Cambridge University, UK *Queen Mary University of London, UK

Content

- Background - from content to service caching.
- Motivation - why do we need service caching?
- SCANDEX - behaviors, key components and etc.
- Challenges - service abstraction, synchronisation and etc.
- Conclusion

From Static Content to Dynamic Service

- Internet is for disseminating information efficiently.
- Our network architecture originates from P-to-P paradigm.
- What should be the next move on research agenda?
  - Information should **not** only refer to static content.
  - Recursive definition: Information = f(Information).
  - f is service which filters, edits, combines existing information to provide new information.
- Ideal Internet should be Information-Centric + Delay-Tolerant.

UNIVERSITY OF CAMBRIDGE

Real-life Needs for Service Caching

- Better localised communication: latency, bandwidth, availability …
- Better control on sharing conventional static content.
- Flexible policy configuration but with simpler architecture.
- Key services in emergency and disaster scenarios.
- Efficient access to popular Internet cloud-based services.

Let’s “scale up” services at network edge.

Communications in Developing Regions

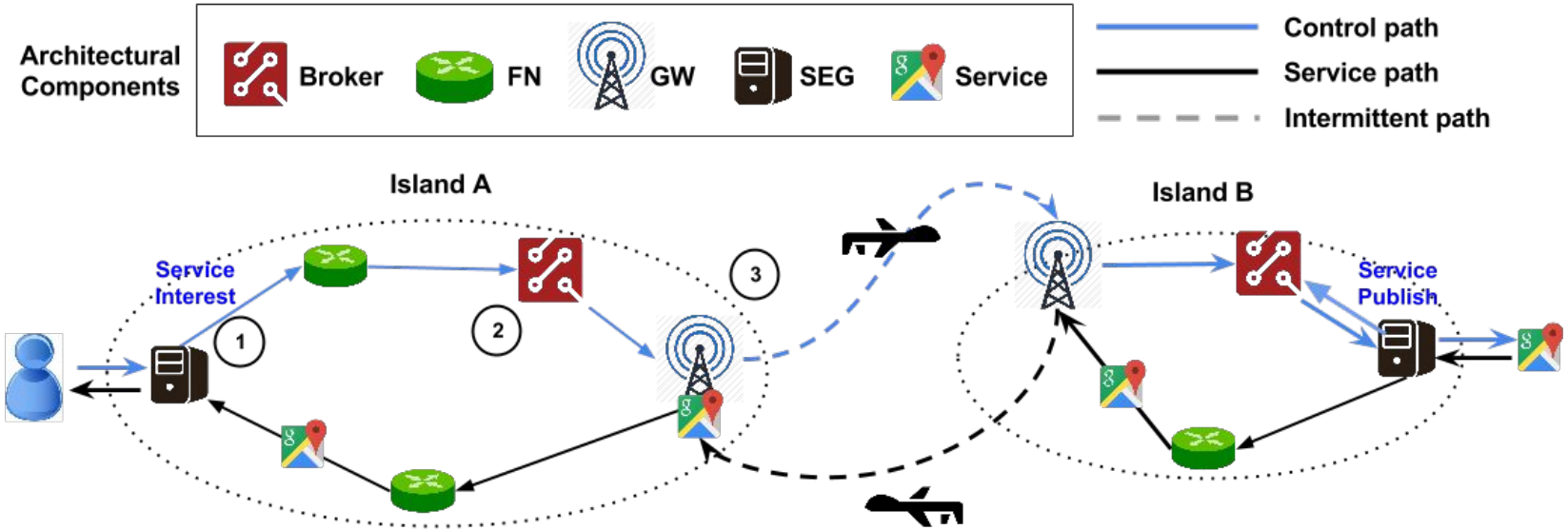
- Poor communication infrastructures.
- Unreliable networks due to natural and technological causes.
- Intermittent connections, not only to end users.
- High cost to access backbone networks.
- Limited coverage vs. widespread population and demands.

SCANDEX - Key System Components

- Service Execution Gateway - executes services.
- Forwarding Node - forwards requests and caches services.
- Edge Gateway - interconnects different domains.
- Broker - service registration and resolution.
- Note multiple components can be multiplexed on one node.

A Utopia for Service-Centric Networking

- SCANDEX depicts a utopia of SCN.



UNIVERSITY OF CAMBRIDGE

System Features

- SCANDEX is a strawman architecture which describes how the system should behave instead of defining protocols.
- Everything is a service, including static content.
- System supports multiple transmission strategies: IP, DTN …
- System intelligently chooses the most suitable strategy.

Questions Instead of Solutions

E.g., how to handle dynamics, predictability, cost, scalability, audit, and etc. More specifically,

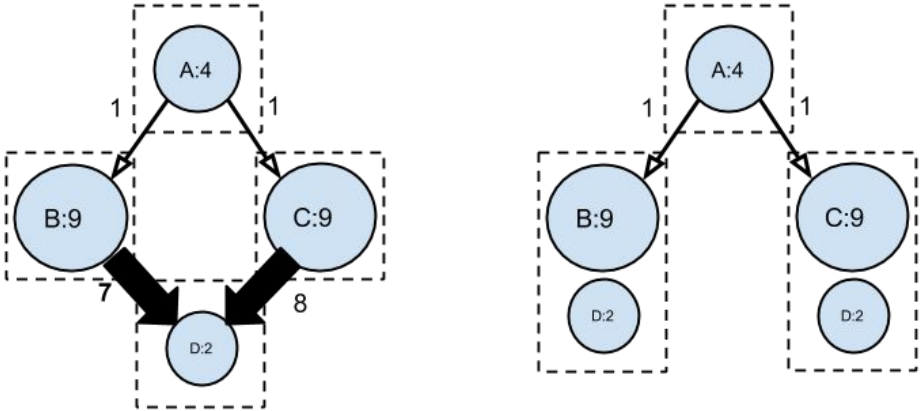
- How to define edge? Network boxes or end-user devices?
- How to differentiate from NFV (network func virtualisation)?
- What naming scheme to use? Flat or hierarchical?
- How to do service resolution? I.e., registration & discovery.

Design Challenges - Synchronisation

- Abstraction: algo + data. In practice, what is it?
- Classification: computation vs. data intensive.
- Caching: singleton vs. multi-instance.
- Synchronisation: stateless vs. stateful.
  - Service-level synchronization semantics.
  - System is only responsible for coordination.

Design Challenges - Dependency

- A hard decision → atomic & meta vs. composition.
- Dependency is modelled as a DAG.
  - How many types of cost? Computation, traffic and etc.
  - Who should resolve the dependency?



UNIVERSITY OF CAMBRIDGE

Other Challenges

- Transmission strategy selection - topology- and context-aware?
- Distributed authentication - connection- or service-based?
- Service instantiation - where and how?
- Caching strategy - proactive or passive?

Summary

- Localised communication requires us to shift from content caching to service-caching.
- SCANDEX defines the high-level system behaviors.
- We defined the key system components and their functions.
- We presented design challenges such service abstraction, synchronisation, dependency, caching etc., and discussed the possible solutions

Thank you. Questions?

UNIVERSITY OF CAMBRIDGE

More Technical challenges

- From content to service - a ‘small’ but non-trivial step.
  - Which services should be migrated?
  - When should they be migrated?
  - Can cached services continue to operate without remote connectivity?
  - How should state be managed?