Design and Realization of an Information-centric Networking Architecture

Dirk Trossen, Computer Laboratory
Information at the Heart of the (Future) Internet

• We all know about video
  • Over 4 billion hrs of videos watched on YouTube every month
  • The 2012 Olympics broke all records
    • BBC delivered 2.8 petabytes on its busiest day, 700Gb/s during the B. Wiggins’ gold
  • …in all this, mobile usage just started to take off!

• But there is more than just video
  • Computation (‘the cloud’)
  • Sensors, Internet of Things and all the Smart XXX

Anything is information and information is anything, really?!
That is correct... (to a point to be discussed)

**BUT**: Economics have changed the possible starting points for a design

- Computing and storage resources are NOT scarce anymore; this led to an almost ubiquitous availability of processing and memory
- Information availability has changed attitude of users
  - **WHAT** is primary, **WHO** and **WHERE** mostly secondary!

⇒ **Location loses its meaning!**

⇒ **Creates desire to fully optimize resources** (wherever they are)
A systems approach that operates on graphs of information with a late (as late as possible) binding to a location at which the computation over this graph is going to happen, enables the full potential for optimization!

The solution to this hypothesis lies in the way we assemble systems!
• One wants to solve a **problem**, each of which might require solving another problem

• Examples:
  • Send data from A to B(s), involving fragmentation along the link(s)
  • Disseminate a video over a local network

• Problems involve “a *collection of information that*” an implementation “*can use to decide what to do*”, which is to implement a problem solution (*)

-> Network architecture, supporting computation in distributed systems is all about *information dissemination* (pertaining to a task at hand)

Starting from Desired System Properties…

• Manipulation of (structured) information flows for computational purposes
  • Expose service model and provide late binding (*WHAT-*WHO*)
• Modularity within a single computational problem
  • Provide modular core functions (*enable optimization*)
• Modularity across computational problems
  • Provide rigorous but flexible layering (*deconstrain constraints*)

...Translated into Design Tenets...

• Provide means for identifying individual information (items)
  • Can be done via labeling or naming (latter requires mapping mechanism)

• Provide means for scoping information
  • Allows for forming DAGs (directed acyclic graphs) of information

• Expose service model
  • Can be pub/sub

• Expose core functions
  • Rendezvous, topology management, and forwarding

• Common dissemination strategy per sub-structure of information
  • Defines particulars of functional implementation and information governance
The problem in question can be implemented through an assembly of sub-problem solutions, whose individual dissemination strategies are not in conflict with the ones set out by the problem in question.

• Hence, problems are assembled to larger solutions by recursively applying the scoping tenet!

• Conflicts are avoided through design and re-design, e.g., via standards procedures!

• Can extend this to runtime reconciliation!
...And Placed into a Layered Model

Layer n

Layer n+1

Layer n-1

Information flow manipulation

Deconstraining through recursive layering

Dissemination Strategy

Rendezvous

Topology

Forwarding

Problem-specific operations

Optimization through modularity within each problem

The layering process is recursive(*)!

Coming Together in A Global Architecture

Node Architecture
- Apps: pub, sub
- ITF

Service Model
- Topology: ITF
- Rendezvous: RP
- Caching
- Helper: ...
- Error Ctrl

Network Architecture
- TM
- FN

Forwarding
- Forwarding Network
Prototype, Deployment & Some Results

Making it work and run - where have we gotten to?
Our Prototype: Blackadder

- Implements design tenets
- Based on Click platform (*)
  - Easy user/kernel space support
  - Easy porting onto other OSes
  - Easy plugging into ns-3
- Domain-local throughput reaches 1GB/s
- Available as open source at https://github.com/fp7-pursuit/blackadder

Our Test Beds

- 9 international sites
- 26 machines with +40 on-demand ones
- tunneled via openVPN for configurable topologies

Also available:
- Dedicated 1GB/s test bed with 15 nodes
- Planetlab (>100 nodes)
- Emulated topologies via ns-3
Experimental Evaluation: Fast Path

Forwarding efficiency

- 15 in a chain
- Multicasting (when nodes is sub)
- ~line speed even when 3 subs per node for 13 nodes
- Degradation when 6 pubs and more due to local copies
Experimental Evaluation: Slow Path

100,000 adverts under single scope

- Subscribers subscribe to random item, wait until receive it and reiterate (500 times)

- worst case for slow path (ignores any possible optimizations due to domain-local rendezvous or mutable semantics)

Node-local
- No net delays
- No TM
- 20ms for 500 processes

Domain-local (Gbit LAN)
- Centralized TM
- ~400 ms for 500 processes per node (7000 subscribers)

Domain-local (PlanetLab)
- Large delays
- ~200ms for 1 sub per node (73 in total)
- ~680ms for 36,500 subs
Comparison with CCN(x)

Node-local (payload size: 2 & 100KB)
- CCNx application expresses interest for 10000 items (/content/segmentNumber)
- CCNx replays all data from the local content store (to avoid signing penalty)

Domain-local (13 nodes in 1GB/s)
- Realize simple window-based flow control
- CCNx replays all data from the first hop cache (to avoid signing penalty)
  - Throughput falls to 170KB/sec if signing each packet on the fly!

Significantly Larger Throughput!
Use Case: Media Delivery

• Media producers increasingly have to deal with new legal and ethical issues.
  • Who can watch our content?
  • When can our content be shown or is legal?
  • What are our responsibilities if our content is shown somewhere where it is illegal?
• In the current internet, it is easy for any user to access any content – which puts media producers at risk.
• So how can media producer maximise their viewers without breaking the law around the world?
A Stakeholder Use Case

- CTVC has been commissioned to produce a film for a UK broadcaster
- CTVC sells the rights to the film internationally
- During production CTVC films its own footage of interviews etc.
- CTVC also sources third party content from libraries
- However, some of the content from the libraries cannot be shown in international territories…
- …so CTVC must make several versions of the same film.
How ICN Could Help…

- Disintegration of content:
  - clips can be pulled from anywhere in the network and re-assembled at the subscriber.

- Encapsulation of media content:
  - when you add new media content with n versions, you only need to add n EDLs (~2kb file), as opposed to n videos (++Mbs/Gbs).

- Opportunistic caching:
  - Once clips are created (expensive), they too can be cached, further optimising the subscription process.

- Complexity of Middleware solutions:
  - Lowers the complexity of the required middleware and, therefore, the complexity of applications (like the SDN!).
Querying for Media: Not Just a Mock-Up!

Pursuit...

HD subscription

Video...

over 18

query

SDN (Player)

UCAM Summer School Promo

(EDL)

Middleware/Blackadder/Publishers
How the Middleware can help…

• By tagging the media with metadata, we can enforce conditions;
  • content marked “adult only” cannot be accessed by users under 18.
  • content tagged “not Saudi Arabia” cannot be accessed in Saudi Arabia.
  • Content tagged “after 8pm” will not work before 8pm depending on the user’s time zone.
  • Clip metadata cannot contradict programme metadata. (You won’t get a “Over_18” clip in a cartoon!)

• How can we achieve this?
  • Ontologies & Reasoners!
What is the Take-Away Here?

- Information-centric networking is **NOT** about disseminating information *because the Internet is no good at it!*

- Information-centric networking is about utilizing the entire design space provided by information (& storage) as well as computation
  - Aided by technological developments that made computation (and storage) ubiquitously available

- To get there, we need to re-think how we design/build systems
  - We have first results, working prototypes, and a growing test bed
What is the Take-Away Here?

• Main principles of our work
  • Center around information
  • Allow for flexible graph-based information structures with semantic-free labels as identifiers
  • Clearly separate functions for finding information, creating appropriate delivery relations, delivering

• Node design for making it all work
  • Running & performing!

• Test beds for making it all real
My Closing Words (for now)

• Please come to http://www.fp7-pursuit.eu to understand more!

• Please download Blackadder from https://github.com/fp7-pursuit/blackadder to play with it!

• Please send me an email at dirk.trossen@cl.cam.ac.uk with any questions or suggestions!