Infomax: An Auto-summarizing Information Delivery Protocol

Tarek Abdelzaher
University of Illinois at Urbana Champaign
The Application Landscape

What trends characterize big future shifts in the application landscape?
The Age of Data

Ubiquitous digital sources:
- Cell-phones
- Glucose monitor
- Smart Meter
- Wii
- Pulse oximeter
- GPS
- Smart Eyewear
- Smart Sportsware
- Smart Watch
- Kinect
The Age of (Democratized) Broadcast

- Ubiquitous digital sources:
  - Wii
  - Glucose monitor
  - Cell-phones
  - Smart Meter
  - Pulse oximeter
  - GPS
  - Smart Eyewear
  - Kinect
  - Smart Sportsware
  - Smart Watch

- Unprecedented dissemination opportunities

YouTube
facebook
flickr
twitter
The Age of (Democratized) Broadcast

- Broadcast in the 20\textsuperscript{th} century:
The Age of (Democratized) Broadcast

- Broadcast in the 20th century:

- Broadcast today:
The rate of data production will increasingly outpace application data consumption needs
A Paradigm Shift
Information Retrieval → Information Distillation

Matching Data at Sources

TCP (Transport Control)
1 bit sent → 1 bit received

Data Received at Sinks

Current Applications
(HTTP, FTP, VoIP, etc.)
A Paradigm Shift
Information Retrieval → Information Distillation

Matching Data at Sources

TCP (Transport Control)
1 bit sent → 1 bit received

Data Received at Sinks

Current Applications (HTTP, FTP, VoIP, etc.)

Matching Data at Sources

Reduction, subsampling, summarization, …

Data Received at Sinks

Future Applications
A Paradigm Shift
Information Retrieval →
Information Distillation

Matching Data
at Sources

In NDN, the network is aware of
(i) application-level object boundaries and
(ii) topological relations between object names.

This knowledge can be exploited for data volume reduction

Data Received
at Sinks

Reduction, subsampling, summarization, …

Future Applications
A “Transport Layer” Solution

Infomax
Infomax: A Novel Data Retrieval API

- Get (/subtree)

- Semantics:
  - Retrieve a representative sampling of data objects under /subtree
Infomax:
A Novel Data Retrieval API

- Get (/subtree)

- Semantics:
  - Retrieve *a representative sampling* of data objects under /subtree
    - Note 1: Representative sampling → minimally redundant
Infomax:
A Novel Data Retrieval API

- Get (/subtree)

Semantics:
- Retrieve a representative sampling of data objects under /subtree
  - Note 1: Representative sampling → minimally redundant
Infomax:
A Novel Data Retrieval API

- Get (/subtree)

- Semantics:
  - Retrieve a representative sampling of data objects under /subtree
    - Note 1: Representative sampling
      → minimally redundant
    - Note 2: Longer shared prefix between objects
      → more semantic redundancy
A Content Transmission Prioritization Policy

Least shared prefix first
A Content Transmission Prioritization Policy

Least shared prefix first

Tie? Take leftmost branch
A Content Transmission Prioritization Policy

Least shared prefix first

Tie? Take leftmost branch

Note: Reduces approximately a breadth-first traversal of the content under /subtree
A Content Transmission Prioritization Policy

Least shared prefix first

Tie? Take leftmost branch

Note: Reduces approximately a breadth-first traversal of the content under /subtree
A Content Transmission Prioritization Policy

Least shared prefix first

Tie? Take leftmost branch

Note: Reduces approximately a breadth-first traversal of the content under /subtree

/subtree

1

2
A Content Transmission Prioritization Policy

Least shared prefix first

Tie? Take leftmost branch

Note: Reduces approximately a breadth-first traversal of the content under /subtree
A Content Transmission Prioritization Policy

Least shared prefix first

Tie? Take leftmost branch

Note: Reduces approximately a breadth-first traversal of the content under /subtree
Protocol Design

**Consumer**

**Get** (/\textit{subtree})
Requests all content under \textit{subtree}

Request objects on \textit{list} in order (fewer requests = more summarization)

**Producer** (advertises /\textit{subtree})

Send interest packet for "subtree/_list/1"

Send data object "subtree/_list/1"

Send interests in listed names

Send data packets

Send interest packets for next "list" (e.g., subtree/_list/2)

Send data object "subtree/_list/2"

Repeat until the consumer gets enough data
Evaluation

- Currently, evaluation is underway on testbed.