Coupling Caching and Forwarding: Benefits, Analysis & Implementation

http://www.enst.fr/~drossi/ccnSim

Dario Rossi
dario.rossi@enst.fr

Giuseppe Rossini
giuseppe.rossini@enst.fr
• **Information centric networks**
  – Mobility, naming, security, …, and Caching

• **Caching on Information centric networks**
  – Not a goal on its own, but a means (e.g., congestion-aware to reduce load on bottleneck links, or cost-aware to reduce ISPs opex)
  – General viewpoint: Efficiently allocate/use caching resources

• **Important notions**
  – Temporary vs Persistent
  – On-path vs Off-path

• **This talk**
  – How to efficiently find the closest, possibly off-path, content replica
Terminology

ICN design space is wide

- We may use FIB toward custodian if available
- We ignore proactive/periodic dissemination of cached copies availability on control plane
- We focus on reactive discovery of cached copies on the data plane

Algorithmically speaking, a cache network can be identified as a triple $\langle F, D, R \rangle$.

- $F$ is the Forwarding strategy
- $D$ is the cache Decision policy
- $R$ is the cache Replacement policy
Related work

- Typically, focus on either $F$ or $D$, but not both ($R=\text{LRU}$ as replacement policy makes sense)

<table>
<thead>
<tr>
<th>Meta-caching $\mathcal{D}$</th>
<th>Type</th>
<th>Knob</th>
<th>Ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fix</td>
<td>Prob.</td>
<td>$p$</td>
<td>[5, 25, 35]</td>
</tr>
<tr>
<td>ProbCache</td>
<td>Prob.</td>
<td>Distance</td>
<td>[29]</td>
</tr>
<tr>
<td>LCD</td>
<td>Det.</td>
<td>Distance</td>
<td>[24, 25, 35]</td>
</tr>
<tr>
<td>WAVE</td>
<td>Prob.</td>
<td>Distance</td>
<td>[13]</td>
</tr>
<tr>
<td>Btw</td>
<td>Det.</td>
<td>Centrality</td>
<td>[9]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Forwarding $\mathcal{F}$</th>
<th>Type</th>
<th>Knob</th>
<th>Ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source routing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flooding</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INFORM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CATT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NDN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>iNRR</td>
<td>Det.</td>
<td></td>
<td>[6, 10]</td>
</tr>
</tbody>
</table>

$F=\text{SPR}$ by default

$D=\text{LCE}$ by default
Agenda

• Benefits
  – Interest of joint <F,D>
  – Comparison with several CDN strategies [16]

• Analysis
  – Modeling of <iNRR,LCE,LRU>
  – Extend <SPR,LCE,LRU>, aka aNET [31]

• Implementation
  – Notion of “Meta-interest” to avoid pollution across paths
  – Spoiler: Arbitrarily close to iNRR, slight delay tradeoff


Coupling Caching and Forwarding: Benefits, Analysis & Implementation

(scenario details in the paper)
(scenario scripts available at)
http://www.enst.fr/~drossi/ccnSim

Part I
CDN vs ICN scenario

Fig. 1. Redundant 4-level binary tree. Dashed links are present with probability $\mu$. Shadowed blocks represent aggregate caches seen by lower level nodes in presence of redundancy ($\mu > 0$).
CDN Policies

- **Edge**: Naive CDN, with all caches operating independently
- **EdgeCoop**: Smarter CDN, all caches cooperate (with iNRR)
- **EdgeNormCoop**: As EdgeCoop, but individual caches have double size (network-wide same budget w.r.t. ICN scope)
ICN Policies

<SPR,LCE>  Naive ICN: on-path caching toward repository
<NRR,LCE>  Access to off-path copies, but cache pollution due to LCE
<NRR,LCD>  Smarter ICN: off-path copies, limits cache pollution via LCD

Configurations:  2-levels vs Ubiquitous caches
(as for EdgeNormCoop, keeping the same overall cache budget)
CDN vs ICN policies

Average distance [hop]

Redundancy probability $\mu$

naive CDN
on-path ICN
smart CDN
naive off-path ICN
smart CDN
smart off-path ICN

2 Levels
Ubiquitous

Edge
EdgeCoop
EdgeNormCoop

<iNRR,LCE>
<iNRR,LCD>
CDN vs ICN policies

Average distance [hop]

Redundancy probability $\mu$

naive CDN

on-path ICN

smart CDN

naive off-path ICN

smart CDN

smart off-path ICN

Edge

$<$SPR,LCE$>$

EdgeCoop

$<$iNRR,LCE$>$

EdgeNormCoop

$<$iNRR,LCD$>$

2 Levels

Ubiquitous

Edge

$<$SPR,LCE$>$

EdgeCoop

$<$iNRR,LCE$>$

EdgeNormCoop

$<$iNRR,LCD$>$

2 Levels

Ubiquitous

Telecom ParisTech
Remark #1

Based on this difference [16], dismisses ubiquitous caching, arguing that most of the gain is achievable painlessly at the edge.
CDN vs ICN policies

Remark #2

Our results are conservative as in [16] <iNRR,LCE> is better than EdgeNormCoop
**Remark #3**

Benefits of ICN appear only
- with meta-caching
- with enough path diversity

**Remark #4**

Ubiquitous caching exhibits
- incremental benefits (aggregation)
- diminishing returns (popularity skew)
CDN vs ICN policies

Remark #5

- Technico economic studies needed to assess if benefits justify deployment
- ICN success unlikely driven only by caching benefits

Technico economic studies needed to assess if benefits justify deployment
Coupling Caching and Forwarding: Benefits, Analysis & Implementation
Modeling iNRR

• The Ideal Nearest Replica Routing (iNRR) [2] is nice and worth *modeling*

• Let start from the aNET model [31] for SPR

\[
  r_{i,v} = \lambda_{i,v} + \sum_{u : R(u, S(i)) = v} m_{i,u} 
\]  

\[
  p_{i,v} = \frac{r_{i,v}}{\sum_{j=1}^{N} r_{j,v}} 
\]  

\[
  \pi_v = LRU(\bar{p}_v, |v|) 
\]  

\[
  m_{i,v} = r_{i,v}(1 - \pi_{i,v}) 
\]
Modeling iNRR

\[ r_{i,v} = \lambda_{i,v} + \sum_{u:u \in N(v)} m_{i,u,v} \]  \hspace{0.5cm} (5)

\[ p_{i,v} = \frac{r_{i,v}}{\sum_{j=1}^{N} r_{j,v}} \]  \hspace{0.5cm} (6)

\[ \tilde{\pi}_v = LRU(\vec{p}_v, |v|) \]  \hspace{0.5cm} (7)

\[ m_{i,v} = r_{i,v}(1 - \pi_{i,v}) \]  \hspace{0.5cm} (8)

Define split ratio among multiple paths

\[ s_{i,v,u} = \sum_{x:R(v,x)=u \land x \in B(v,S)} \left[ \prod_{y \in B_i(v,x)} (1 - \pi_{i,y}) \right] \frac{\pi_{i,x}^2}{\sum_{z \in B_b(v,x)} \pi_{i,z}} \]  \hspace{0.5cm} (9)

Applies split ratio to miss stream

\[ m_{i,v,u} = \begin{cases} m_{i,v} s_{i,v,u} & u \neq R(v, S) \\ m_{i,v}(1 - \sum_{w \neq u} s_{i,v,w}) & u = R(v, S) \end{cases} \]  \hspace{0.5cm} (10)

Similar to aNET
Model accuracy

aNET suffers from IRM violation due to miss-stream far from the leafs

LRU approximation very accurate for leaf nodes

iNRR shorter paths and larger fanout, make IRM violation effect less pronounced

Fig. 7. Scatter plot of the average cache hit per node $\bar{\pi}_v$ obtained via simulation vs model, for aNET and iNRR, on a 10x10 grid.
Coupling Caching and Forwarding: Benefits, Analysis & Implementation

Part III
Architecture

- General scheme: forward interests along either
  - FIB for persistent on-path copies, proactively advertised by routing
  - TFIB for temporary off-path cached copies, reactively discovered via exploration

Explore

Interests

Data

Exploit

TFIB=

FIB=
x:3

TFIB={x:1}

FIB={x:3}

TFIB={}

FIB={x:3}

1 3 5

153

1

1

3

time
The Ideal Nearest Replica Routing (iNRR) [2] is nice and worth *implementing*

- Explore via scoped flooding, at distance $d$ hops

1-phase design (NRR’)

*Use regular interest packets*

All cache hits return real data. Redundant for popular content, useless for unpopular content

2-phase design (NRR’’)

*Use meta-interest packets*

Hits indicate data availability. Afterward, actual requests only sent through a single TFIB path
Practical NRR very close to iNRR

1-phase
Regular interests yield to cache pollution penalty

2-phases
Meta-interests confine pollution to a single path

Meta-caching limit pollution along the path

Meta-interests + LCD meta-caching = \langle NRR'', LCD \rangle \text{ arbitrarily close to iNRR}
Implementing NRR

- Implementing ideal Nearest Replica Routing (iNRR)
- Explore via scoped flooding, max distance $d$ hops

1-phase design (NRR’)

*Use regular interest packets*

Regular interest generate cache decision, triggering cache replacement and pollution

Fastest propagation, time to first chunk is lowest

2-phase design (NRR’’)

*Use meta-interest packets*

Meta-interest return only data-availability indication, not data itself. Avoid Pollution!

Slower propagation, time to first chunk longer (subsequent chunks are found at NRR)
Implementing NRR

• Implementing ideal Nearest Replica Routing (iNRR)
• Explore via scoped flooding, max distance $d$ hops

In practice, delay penalty
• only affects the first chunk, upper-bounded by network RTT

Delay not necessarily longer
• as content is closer, 1st chunk delay may even be shorter

Fastest propagation, time to first chunk is lowest

Slower propagation, time to first chunk longer (subsequent chunks are found at NRR)
Coupling Caching and Forwarding: Benefits, Analysis & Implementation

Aftermath
Conclusions

• Contributions
  – Benefits, analysis and implementation of joint forwarding and caching
  – *Meta-caching*: implicit coordination of caching decisions *along a single path*
  – *Meta-interests*: implicit coordination of caching decisions *across multiple paths*

• Limits
  – Can we do better (e.g., 2-LRU instead of LCD) ?
  – How far are is <iNRR,LCD,LRU> from optimum ?
  – Only LCE is modeled
  – Only synthetic topologies are simulated, with stationary popularity and no spatial skew
Implications

• Comparison guidelines
  – Algorithmic suggestion for IRTF ICNRG baseline comparison
  – Good practice = include <iNRR,LCD,LRU> (or better)
  – Bad practice = using only naive <SPR,LCE,LRU>

• Comparison is feasible
  – NRR” implementation available in existing tools (e.g., ccnSim)
  – iNRR (and NRR’’) simple to implement in other simulators
  – Results consistent across simulators: Come see the demo!
Backup slides
Interest of joint $\langle F,D,(R=LRU)\rangle$

Some gain
Larger gain

<SPR,LCE>
<iNRR,LCE>

<SPR,LCE>
<iNRR,LCD>

Fig. 2. $\langle F,D \rangle$ performance at a glance: average content distance as a function of meta-caching policies, for SPR (left) and iNRR (right) forwarding, on tree (top) and grid (bottom) topologies.
More realistic catalog/cache scales