MIRCC: Multipath-aware ICN Rate-based Congestion Control

Milad Mahdian
Northeastern University

Jim Gibson
Cisco Systems

Somaya Arianfar
Cisco Systems

Dave Oran
Cisco Systems
Outline

I. Introduction
II. Single-Path MIRCC
III. Multipath MIRCC
   I. Dual-Class Best-Subflow Multipath Rate Management
IV. Multipath Evaluation
V. Conclusions
I. Introduction
Why Should ICN Explore Rate-Based Congestion Control?

• Many attractive elements, in spite of lack of take-up under IP
  – N. Dukkipati, “Rate control protocol (rcp): Congestion control to make flows complete quickly”, 2008
  – Quick determination of sending rate (from first packet received)
  – Continuous forcing of full buffers to track congestion point not required
  – Max-min fair allocation of bottleneck links between competing flows
• ICN differs from IP in ways that may be well-suited to rate-based congestion control
  – Receiver-based with Interest/Data balance
  – Symmetric forwarding
• ICN relatively clean-slate
  – IP’s window-based congestion control has assumptions and expected behavior
  – Compliance with those assumptions handicaps alternatives and adaptations
Goals and Context

• Provide fairness between flows while maximizing network utilization
  – avoiding congestion and excess latency
  – taking advantage of multipath
  – flow defined by application
• Converge rapidly in response to load change while maintaining stable flow allocation under stable load
• Forwarders not required to maintain long-lived per-flow state
  – Pending Interest Table is short-lived
• Rely on distributed algorithms, i.e.
  – in the style of the current Internet
  – not centralized omniscient controllers
• Caching
  – Multi-destination supported: object-level caching a form of multi-destination
  – Per-chunk opportunistic caching not considered at this time
II. Single-Path MIRCC
Basic Single-Path Rate Control Concepts

• Each forwarder, at each interface, maintains $R(t)$: the per-flow stamping rate for that interface
• Forwarder stamps each Data msg with its interface $R(t)$ if $R(t) < \text{received_value}$
• Thus, each Data message carries lowest $R(t)$ seen along path
• Endpoint consumer updates flow’s Interest sending rate
  • according to Date rate indicated by received bottleneck $R(t)$, deflated
• Forwarder manages load by changing $R(t)$ +/- when observed traffic is low/high
  • $R(t)$ calculation is the heart of the distributed system
Forwarder’s $R(t)$ Elements

- $R(t)$ represents Data rate but is calculated based on Interest rate. Why not use Data rate?
  - ICN style: ICN is pull-based, which implies managing Interests directly, Data indirectly
  - We assume an Interest shaper on each face
    - Shaping/Nacking Interests filters congestion signals from Data rate
- Maintain an *Inflation* (DataSize/IntSize) estimate
  - Use shaper’s inflation value
- Directly stamp Data messages
  - i.e. stamping Interests and reflecting $R(t)$ back in Data at producer
  - Quicker reporting to consumer
R(t) Algorithm Overview

- MIRCC and RCP R(t) algorithms differ in detail, similar in spirit
- R(t) updated once per interval T, stamped for next interval [t+T]
- R(t) inputs:

<table>
<thead>
<tr>
<th>Interval</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link capacity (leaving headroom, e.g. 95%)</td>
<td>ηC</td>
</tr>
<tr>
<td>Rate calculated for previous interval</td>
<td>R(t-T)</td>
</tr>
<tr>
<td>Interests (inflated) arriving during this interval</td>
<td>y(t)</td>
</tr>
<tr>
<td>Queue depth/RTT estimate</td>
<td>q(t)/d(t)</td>
</tr>
<tr>
<td>Constants (weighted average, tuning parameter)</td>
<td>α, β</td>
</tr>
</tbody>
</table>

- Inputs based on aggregates
- **No** flow-specific state/logic
- **No** forwarder determination of how many actual flows are using the link. $\hat{N}$, aka “Flow estimate” is synthetic, i.e. unrelated to actual flows
Simulation Results
Single-Path MIRCC vs. Straight Port of RCP Algorithm

[Cisco implementation of CCNx1.0, run under ns-3]
III. Multipath MIRCC
Issue 1: Path Identification and Steering

A flow consists of multiple subflows. Each subflow’s path reports its own $R_{sf}(t)$.

- What entity tracks a flow’s multiple subflows and their current $R_{sf}(t)$?
  - *Consumer endpoint* (application/application-library), the only entity given stated goals and assumptions
  - Significant extra responsibility for consumer vs. single-path situation

- How does consumer endpoint identify subflows and per-subflow $R_{sf}(t)$?
  - *Path identifier*, reported in Data message

- How are consumer endpoint’s decisions about per-subflow rates honored for the consumer’s Interests?
  - *Path identifier*, reflected back in Interest as hint

- How are subflows discovered?
  - Interests without path identifiers must be sent initially/periodically
  - Forwarders with multiple next hops choose probabilistically

- Path steering mechanisms have other possible uses, e.g. for ICN ping performance measurement and for ICN traceroute
Issue 2: Multipath Fair Flow Rate

• Simple schemes do not meet the stated goals
  – Send at rate of Best-Subflow: \( \text{Max}(R_{sf}(t)) \)
    • Underutilized network (easily evident with small number of flows)
  – Send at Total per-Subflow rate: \( \Sigma R_{sf}(t) \)
    • Flow-Unfair: Not max-min fair between flows
    • Link-Unfair: Can occupy more than one flow-share of shared links

• Signaling a complete picture of flow’s subflow topology (including joint bottleneck link rates) to consumer or forwarders is not tractable
Multipath Fair Flow Rate Solution

Dual-Class Best-Subflow Multipath Rate Management

• Existing ("Primary") Class
  – Forwarder maintains and stamps $R_p(t)$ in every Data message
  – Primary pool continues to be sized to link capacity: $C$
  – Consumer sends Primary Interests at Best-Subflow rate: $\max(R_p(t))$
  – Generally high level of fairness between flows

• New ("Secondary") Class, for any remaining bandwidth
  – Forwarder also maintains and stamps $R_s(t)$ in every Data message
  – Secondary pool sized to unused primary bandwidth: $C – primary\_traffic(t)$
  – Consumer sends Secondary Interests at Total per-Subflow rate: $\sum R_s(t)$
  – "Soaks up" any remaining available bandwidth

• Forwarders preferentially drop Secondary traffic under congestion
III. Evaluation
MIRCC Multipath: Single Flow Arrival and Departure

Rate management for two consumers, one with multipath

(Note: Flow TotalRate = Primary + Secondary)

95% of Bottleneck Link

R(t) = 15 Mbps
R(t) = 12 Mbps
R(t) = 9 Mbps

15-12-9 Slingshot Topology

Note:
- Joint bottleneck \( l_a \) always fully utilized (C1:TotalRate + C2:TotalRate)
- Stable flow rates under stable load
- While C1 and C2 are competing
  - no excess capacity
  - secondary rate traffic goes to 0
MIRCC Multipath: Mix of Short and Long Flows

Flows arriving/leaving frequently (0.5s)

Note:
- Short flows not disadvantaged
- Links fully utilized
Single-Path vs. Multipath

Note:
- 3 synthetic producer nodes (not really Amazon/Google/Warner)
- 8 consumer nodes, random assignment
- Random mix of flow sizes
- Enabling Multipath significantly lowers average delivery time vs. Single-path
  - not on 14, 15
Conclusions

• We identify an algorithm for maintaining $R(t)$ that provides better throughput and convergence, in our ICN simulations, than (a straight port of) RCP’s algorithm

• For multipath rate-based congestion control
  – We devise a scalable path management solution, operating at the endpoints, that is based on a path identification/steering mechanism
  – We devise a rate management solution that
    • Maintains a high degree of fairness between flows
    • Achieves high network utilization
    • Uses additional machinery that is in the spirit of the simpler single-path rate-based congestion control
  – We avoid forwarder per-flow state.
Q/A

ACM ICN
2016 KYOTO
Appendix. Backup Slides
**MIRCC $R(t)$ Algorithm**

- $R(t) = BaseRate(t) + ExcessRate(t)$
  - the stamping rate
- $BaseRate(t) = \frac{\eta C - \beta(t)q(t)}{\hat{N}d(t)}$
  - non-recursive formula to split the allowed link bandwidth among the passing flows
- $ExcessRate(t) = R(t - T) - \frac{y(t)}{\hat{N}}$
  - recursively fill up the extra available bandwidth with traffic equally
- $\hat{N} = \max(C, y(t)) / R(t - T)$
  - equivalent number of flows with full rate
- $\beta(t) = \max\left( \beta', \frac{y(t) - y(t - T)}{y(t)} \right)$
  - self-tuned parameter chosen for stability
Rate-Based Basic (Single Path) Scheme: Flow Rate Convergence

4 responsive (elastic) flows sharing link
- \( R(t) \) [FlowStampingRate]: 2.5Gig
- FlowCount(Notional): 4

4 responsive (elastic) flows sharing link, 3 are bottlenecked elsewhere at 1G
- \( R(t) \) [FlowStampingRate]: 7Gig
- FlowCount(Notional): 1.43

0 flows currently using link
- \( R(t) \) [FlowStampingRate]: 10Gig (cap)
- FlowCount(Notional): 1/0

100 CBR flows (1 pps) sharing link
- \( R(t) \) [FlowStampingRate]: \( \sim 10 \text{Gig} \) (cap)
- FlowCount(Notional): 1/0
Basic Single-Path Rate Control Properties

- Achieves max-min fairness among flows sharing link
- **No** flow-specific state/logic at forwarder for calculating stamping rate
- *Not* based on forwarder determination of how many actual flows are using the link
- Minimal endpoint responsibilities *in single-path version*
MIRCC Multipath: Single Flow Arrival and Departure
(Symmetric 10Mbps Producer Links)

Note:
- Joint bottleneck $l_a$ always fully utilized
  \( (C1:TotalRate + C2:TotalRate) \)
- Stable flow rates under stable load
- While $C1$ and $C2$ are competing
  - no excess capacity
  - secondary rate traffic goes to 0

Rate management for two consumers, one with multipath

15-10-10 Slingshot Topology

(Producers 1b, 2)