CCTCP: A Scalable Receiver-driven Congestion Control Protocol for Content Centric Networking

Lorenzo Saino, Cosmin Cocora and George Pavlou

Communications and Information Systems Group
Department of Electrical and Electronics Engineering
University College London
{l.saino,c.cocora,g.pavlou}@ee.ucl.ac.uk

Content Centric Networking (CCN)

- Content Centric Networking (CCN)[1] is a recently proposed Internet architecture which shifts the main network abstraction from node identifiers to location-agnostic content identifiers.
- In CCN, content objects are partitioned into addressable chunks which can be contained within a single packet and reactively cached at any router along a path.
- Two network primitives:
  - *Interest* packets, which are routed according to the identifier of the requested content towards the closest available copy
  - *Data* packets, which deliver the requested content chunk in response to an *Interest* packet

[1] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard. Networking named content. ACM CoNEXT 2009

Transport issues in CCN

Since contents are cached with a packet-level granularity, chunks may be served by different network nodes when retrieving an entire content object.

This makes TCP-based implicit-feedback congestion control mechanisms inefficient:

- Out-of-order delivery or variations in inter-arrival times may be caused by adjacent chunks being served by different caches rather than congestion.
- RTO estimation is unreliable because of greater RTT variability caused by frequently changing chunk sources.

Proposed transport protocols for CCN

Currently proposed transport protocols can be categorized in:

- Receiver-driven
  - Control loop in the receiver, stateless routers
  - Proposals: ICTP [2], ICP [3], ConTug [4]
- Hop-by-hop
  - Control loop in the routers which need to keep per-flow state
  - Possibility to control misbehaving receivers
  - Proposals: HoBHIS [5] HR-ICP [6]

---

[2] S. Salsano, A. Detti, M. Cancellieri, M. Pomposini, and N. Blefari-Melazzi, "Transport-layer issues in information centric networks", ICN workshop, ACM SIGCOMM 2012

[3] G. Carofiglio, M. Gallo, and L. Muscariello, "ICP: Design and evaluation of an interest control protocol for content-centric networking", NOMEN workshop, IEEE INFOCOM 2012

[4] S. Arianfar, P. Nikander, L. Eggert, and J. Ott, "Contug: A receiver-driven transport protocol for content-centric networks", in IEEE ICNP 2010 (Poster session)

[5] N. Rozhnova and S. Fdida, "An effective hop-by-hop interest shaping mechanism for CCN communications, NOMEN workshop, IEEE INFOCOM 2012

[6] G. Carofiglio and L. Muscariello, Joint hop-by-hop and receiver-driven interest control protocol for content-centric networks, ICN workshop, ACM SIGCOMM 2012

Content Centric TCP (CCTCP)

Design objectives and assumptions

Design Objectives

- Incremental deployability
- Scalability to Internet scale
- Fairness among CCTCP flows and with legacy TCP flows
- Independence from caching policies and caches location

Assumptions

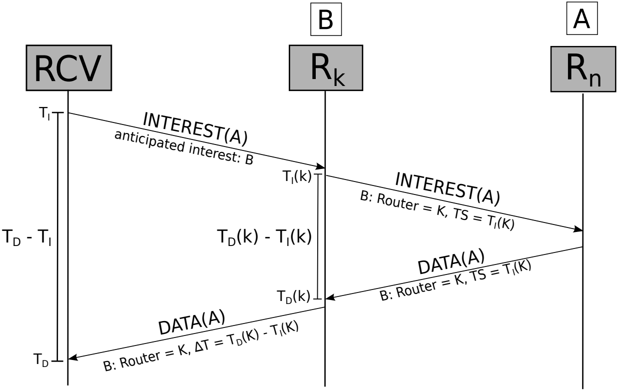
- *Interest* and *Data* packets follow the same path in opposite directions at the CCN layer
- *Interest* packets are routed to the original source. A cached copy is served only if on the path between receiver and original source

Content Centric TCP (CCTCP)

Protocol overview

- Core features:
  - Receiver-driven, window-based, implicit-feedback congestion control
  - *Slow start* and *congestion avoidance* phases based on TCP New Reno
  - Only timeout expiration used as signal of congestion. No window size drop on out-of-order arrivals.
  - Fast recovery at timeout expiration
- CCTCP keeps one retransmission timeout and one congestion window per each expected source of chunks
- Expected source is predicted before sending an *Interest* packet, thanks to the *anticipated interests* mechanism

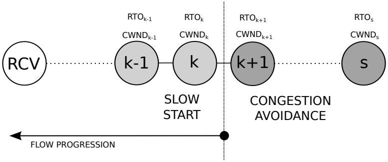
Anticipated interests mechanism



$$RTT(k) = (T_D - T_I) - (T_D(k) - T_I(k))$$

Window and timeout update algorithms

- Timeout calculated according to TCP's Jacobson algorithm
- At each timeout expiration $RTO(i+1) \leftarrow 2 \times RTO(i)$ and $CWND(i+1) \leftarrow CWND(i)/2$
- At each *Data* packet reception, rate is increased for all caches between the receiver and the expected source
- At each timeout expiration, rate is increased for the expected source and all caches beyond it



$$\frac{CWND(k)}{SRTT(k)} \geq \frac{CWND(k+x)}{SRTT(k+x)}$$

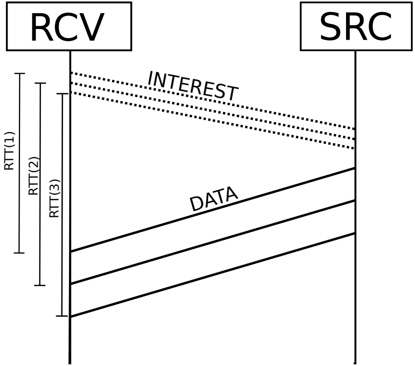
$$RTO(k) \leq RTO(k+x)$$

$$\forall x \in [1, s-k]$$

Pacing of *Interest* packets

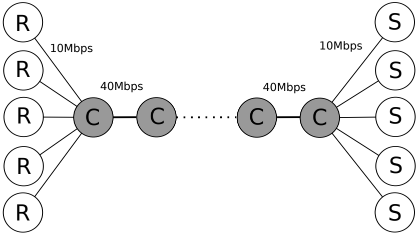
- Data packets are larger than *Interest* packets and therefore also their T_{tx} is larger
- If *Interest* packets are sent in a burts, Data packets sent in response to *Interests* at the end of the burst will experience longer RTT
- Solution: pace the sending of *Interest* packets over an RTT period:

$$T_{send}(i) = \frac{SRTT(i)}{CWND(i)}$$



Performance evaluation

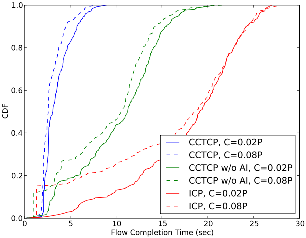
Simulation scenario



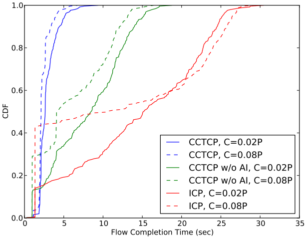
- Metric: Flow Completion Time (FCT)
- Variable conditions:
  - Cache sizes: $0.02P$ and $0.08P$
  - Content popularity α: 0.64, 1.03
  - Caching policies: ALL+LRU, ALL+LFU, RAND+LRU
- Comparison with ICP and CCTCP w/o Anticipated Interests

Performance evaluation

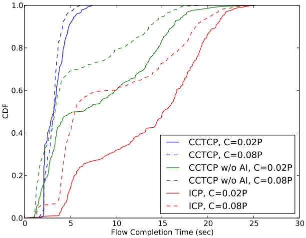
Flow Competion Time (FCT), cumulative probability



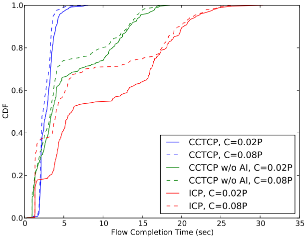
(a) ALL+LRU, $\alpha = 0.64$



(b) ALL+LRU, $\alpha = 1.03$



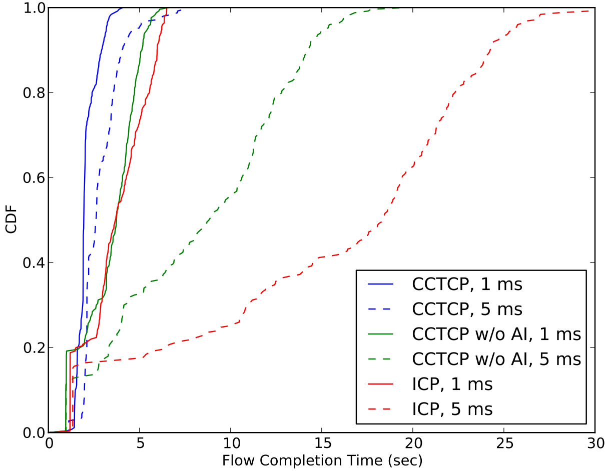
(c) RAND+LRU $\alpha = 0.64$



(d) RAND+LRU, $\alpha = 1.03$

Performance evaluation

Impact of inter-cache distance



Conclusions and future work

- Previously proposed receiver-driven transport protocols for CCN cannot reliably identify congestion by keeping a single timeout value.
- Hop-by-hop protocols can cope with with source variability but at the cost of requiring each router to keep per-flow state.
- CCTCP considerably outperforms other receiver-driven proposals and the performance gain provided by *anticipated interests* is substantial.
- Results presented here have been gathered using an unoptimized algorithm for selecting *anticipated interests* chunks set. Optimized algorithms can further improve performance.