CoAP Simple Congestion Control/Advanced (CoCoA)

draft-bormann-core-cocoa-02

Carsten Bormann – Universität Bremen TZI

cabo@tzi.org

August Betzler, Carles Gomez, Ilker Demirkol

Universitat Politècnica de Catalunya (UPC)/Fundació i2cat

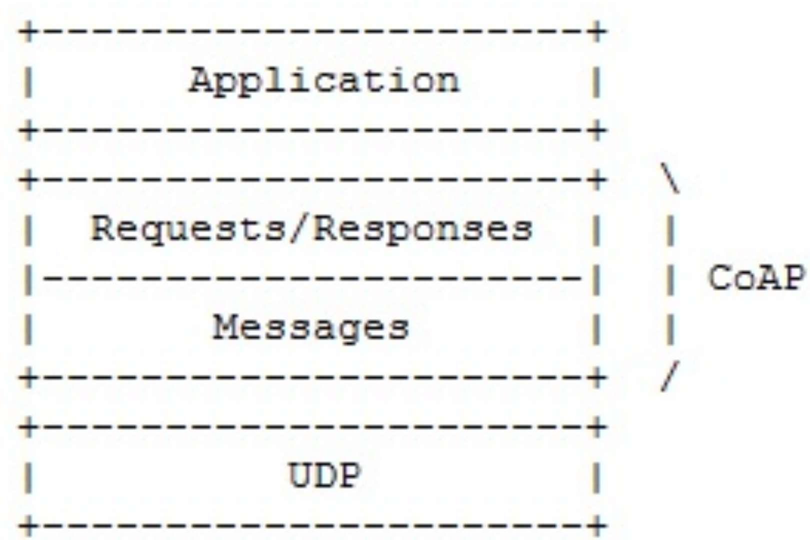
carlesgo@entel.upc.edu

UPC members have been partly supported by the Spanish Government through project TEC2012-32531, and FEDER

IETF 92 – Dallas, March 2015

Context (I)

- Constrained Application Protocol (CoAP)
  - RFC 7252
  - Lightweight, efficient protocol
    - For constrained node networks
  - Over UDP
  - Messages
    - Confirmable (CON)
    - Non-confirmable (NON)



Context (II)

- Default CoAP congestion control for CONs
  - RTO chosen from a fixed interval: [2, 3] s
  - Binary Exponential Backoff (BEB)
  - Outstanding interactions to a destination = 1
- CoCoA
  - Simple mechanism for advanced congestion control
    - Using RTT measurements for CONs
    - Rules for NONs

CoCoA: RTO calculation (I)

- Strong and weak RTTs
  - Weak RTTs: retransmissions have been required
- RTO estimator
  - Input from *weak* and *strong* RTO estimators
  - $RTO_{overall}$ is evolved from the estimator that made the most recent contribution

$RTO_{weak} \leftarrow SRTT_{weak} + 1 * RTTVAR_{weak}$

$RTO_{overall}$

$RTO_{strong} \leftarrow SRTT_{strong} + 4 * RTTVAR_{strong}$

(RFC 6298)

CoCoA: RTO calculation (II)

- Reduced RTO_{weak} contribution:
  - $RTO_{overall} := 0.25*RTO_{weak} + 0.75*RTO_{overall}$
  - $RTO_{overall} := 0.5*RTO_{strong} + 0.5*RTO_{overall}$
  - Only responses obtained before the 3rd retransmission update RTO_{weak}
  - $RTO_{overall}$ is dithered

CoCoA: Variable Backoff Factor

- Goals
  - Avoid too quick retries for low RTO values
    - Could contribute to congestion
  - Reduce too slow retries for large RTO values
    - Could lead to unnecessary delay increase
- Definition
  - RTO < 1 s → VBF = 3
  - 1 ≤ RTO ≤ 3 s → VBF = 2
  - RTO > 3 s → VBF = 1.5

CoCoA: RTO aging

- High RTO values
  - If RTO > 3 s, and not updated for 4*RTO , then
  - RTO = 1 + 0.5*RTO
    - Converge towards default RTO values

- Low RTO values
  - If RTO < 1 s, and not updated for 16*RTO , then
  - RTO = 2*RTO
    - Converge towards default RTO values

Running code

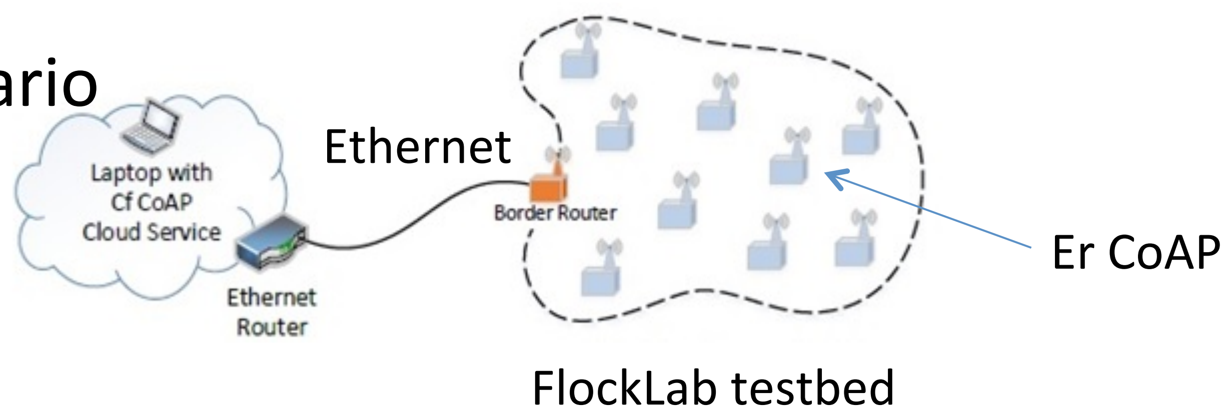
- cocoa-02 has been implemented for Californium (Cf)
  - CoAP implementation for unconstrained platforms
  - Optional *CongestionControlLayer*
- Californium with CoCoA is publicly available
  - https://github.com/eclipse/californium
    - cf-cocoa example
    - org.eclipse.californium.core.network.stack.congestioncontrol
- CoCoA implementation for Erbium (Er) is underway
  - Erbium: official CoAP implementation for Contiki OS

Evaluation: scenarios and results

- Simulation of IEEE 802.15.4 networks
  - With/without reliability, NullRDC/ContikiMAC
  - Various topologies
- Real experiments
  - GPRS scenario

Note: for details, please refer to published/upcoming papers or ask the authors

  - 802.15.4 scenario

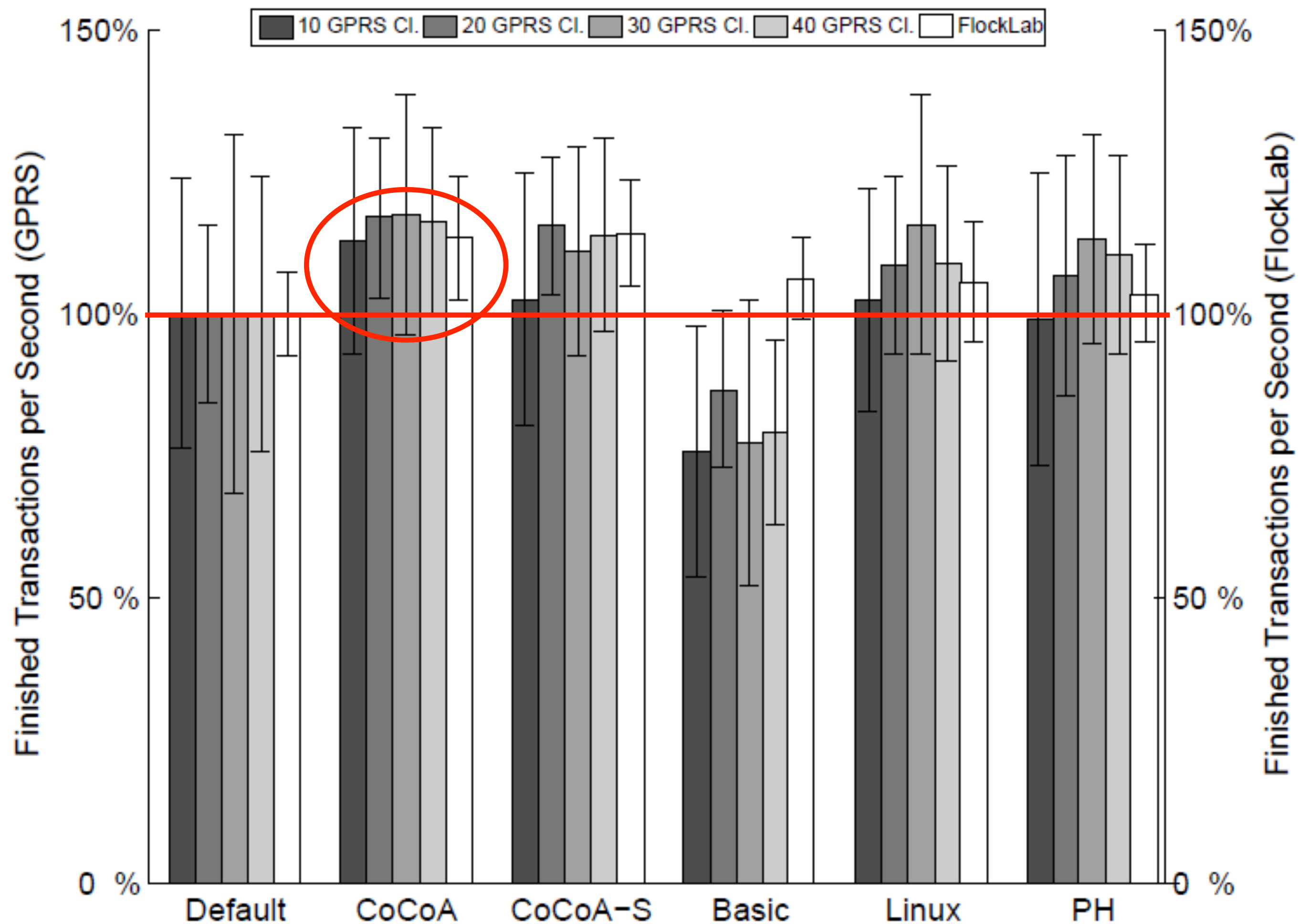


Considered RTO algorithms

- Default CoAP
  - Insensitive to RTT
- CoCoA
- CoCoA-S
  - Strong only
- Basic RTO
  - RTO randomly chosen from [last_RTT, 1.5*last_RTT]
  - Also uses weak RTTs
- Linux RTO
  - Reduces contribution of variance to the RTO when RTT decreases
  - Avoids RFC 2988 RTO getting too close to the RTT
- Peak-Hopper RTO
  - Short history and long history estimator
  - Maximum of the two estimators

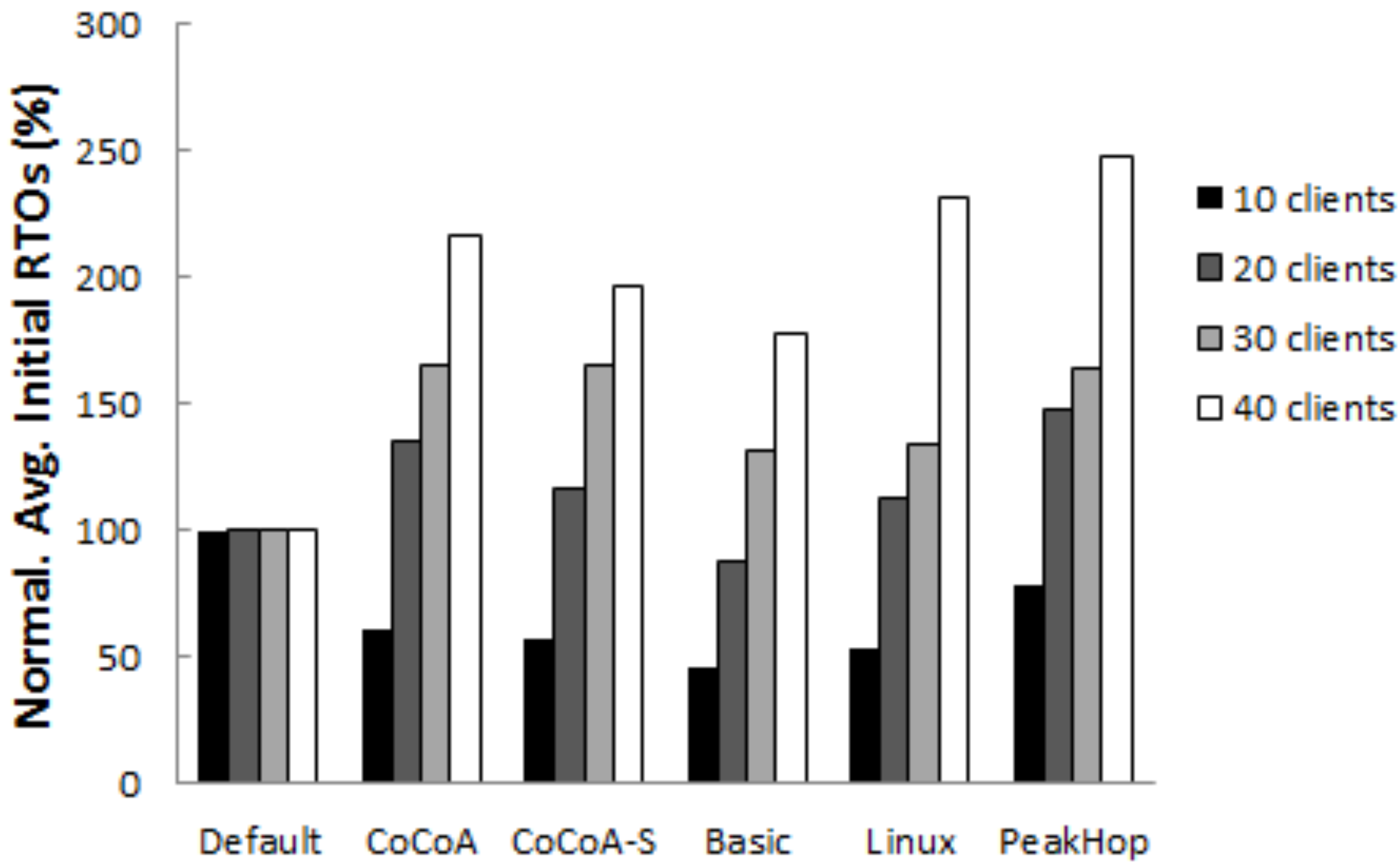
Successful exchanges per time unit

- GPRS and 802.15.4 scenario
  - New CON sent once the previous one is ACKed



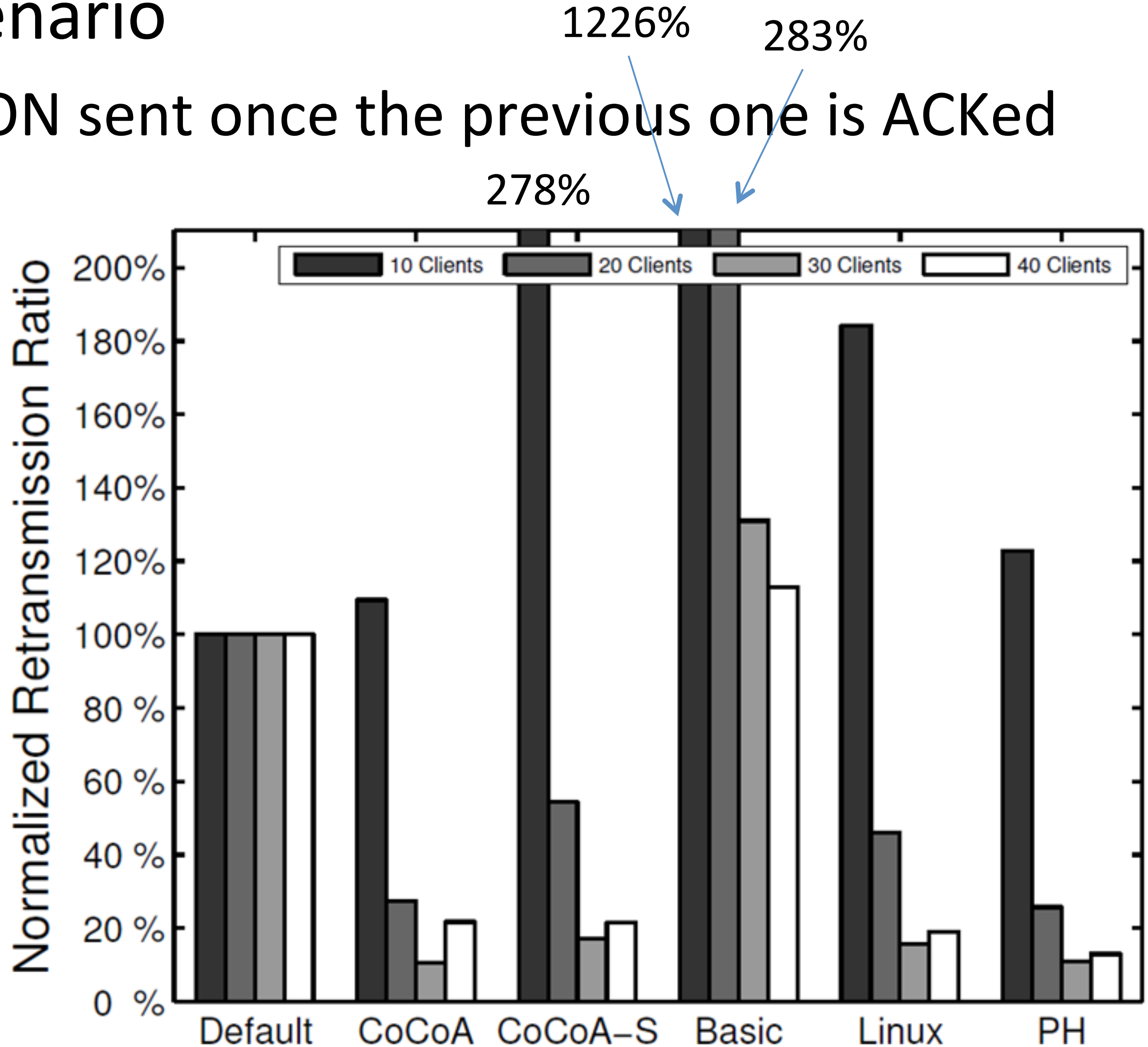
Initial RTO

- GPRS scenario
  - New CON sent once the previous one is ACKed



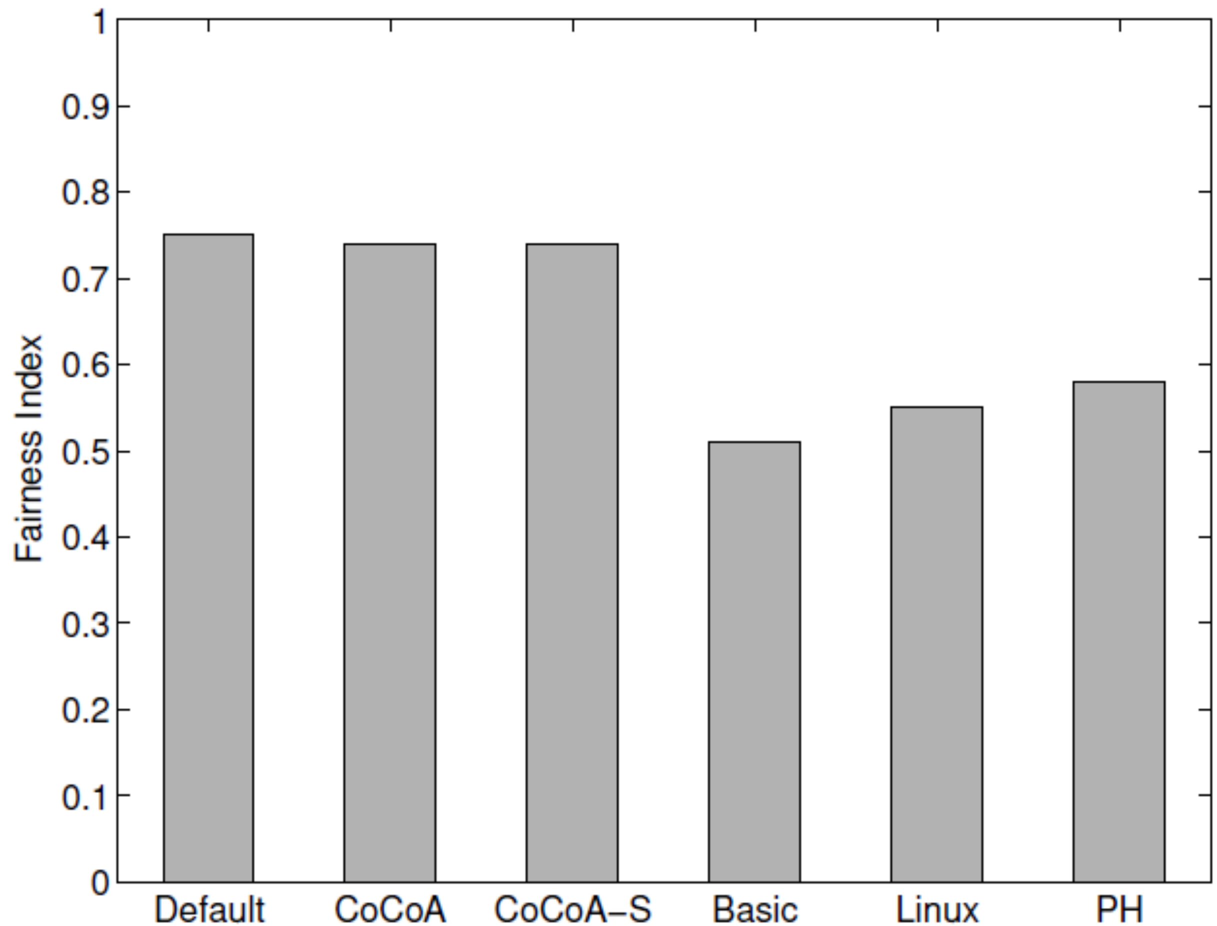
Retry ratio

- GPRS scenario
  – New CON sent once the previous one is ACKed



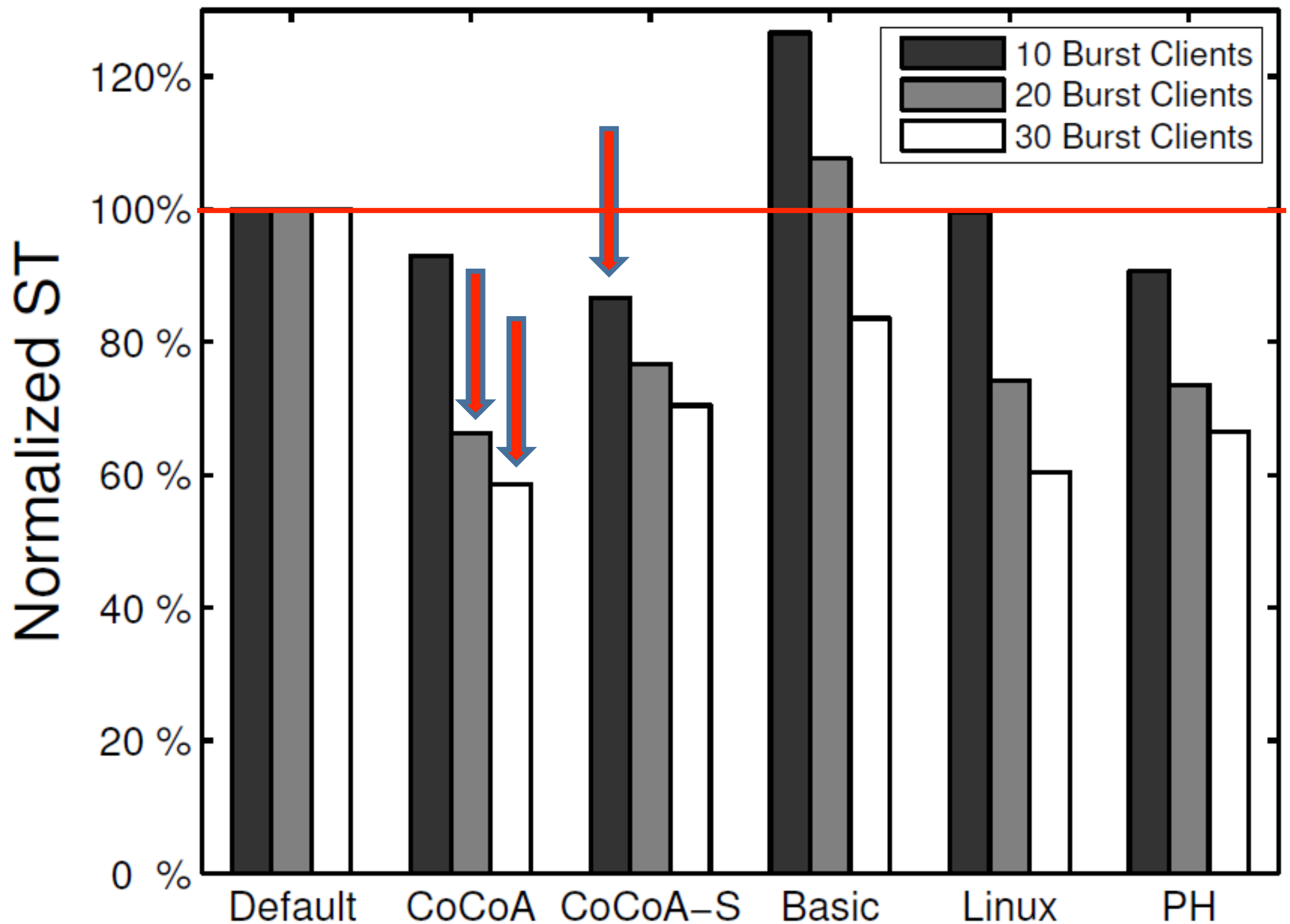
Fairness

- 802.15.4 scenario
  - Fairness index
    - RFC 5166
  - CoCoA does not degrade fairness
    - Variable Backoff Factor
    - Use of weak RTTs



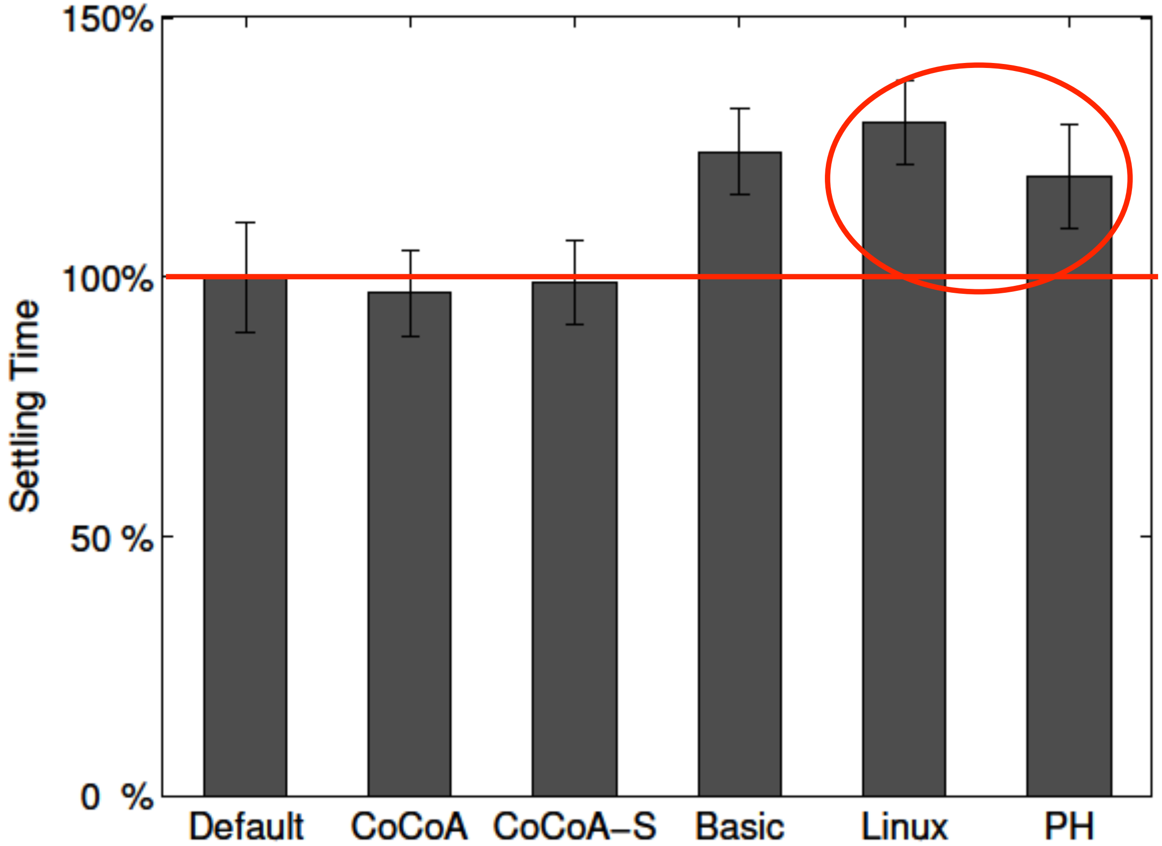
Settling time

- GPRS scenario
  - Time to serve 80% of the requests in a burst



Settling time

- 802.15.4 scenario
  - Time to serve 80% of the requests in a burst
  - TCP-oriented RTO algorithms underperform



Observations (I)

- CoCoA performs similarly to or better than default CoAP
  - Good use of RTT samples
  - Throughput increase, settling time decrease
  - Fairness not degraded
  - Underperformance not observed
- CoCoA-S
  - Often good performance in congested scenarios (vs default CoAP)
    - But high number of retries in low congestion scenario!
  - A bit less conservative than CoCoA
    - No weak RTTs/RTO

Observations (II)

- Too simplistic RTT-sensitive approaches underperform default CoAP
  - Basic RTO considers only the last RTT sample
  - Not enough safety margin (RTO vs actual RTT)
  - Huge amount of (too early) retries

- TCP-oriented RTO algorithms underperform default CoAP in some aspects/scenarios:
  - Weak RTT updates are missed
    - A problem when losses take place
  - Settling time (802.15.4)
  - Fairness (802.15.4)
  - Dithering

Not adapted to IoT scenarios

Memory considerations

- RAM requirements
  - Per client role

	RAM (bytes)
Default CoAP	2
CoCoA	29
CoCoA-S	19
Basic RTO	2
Linux RTO	21
Peak-Hopper RTO	43

Checklist (1/2)

- draft-bormann-core-cc-qq-00
  - Algorithm for general use?
  - Does it protect the network?
    - Compared with default CoAP
    - Regardless of lower layer mechanisms
  - Stable?
    - Synchronization avoided by using dithering
    - Hint on granularity missing
    - RTT history length
    - RFC 6298 behavior and modifications analyzed
  - Scalable?
    - Tested/simulated for networks up to ~ 50 nodes

Checklist (2/2)

- draft-bormann-core-cc-qq-00
  - Range?
    - Higher offered loads needed
    - Low RTT / High RTT evaluated
    - Single-hop / multihop networks evaluated
  - Scope?
    - Possible to consider different destination scopes
    - Aggregate congestion behavior
  - Good performance?
    - Yes (so far…)
  - Fairness?
    - Self-fair
    - Fair with TCP
  - Evaluation quality?
  - Additional security considerations?
    - TBD

Call to Action

- Please implement cocoa-02
  - Is the draft specification clear?
- Please experiment with cocoa-02
  - Performance issues?
  - Improvement possibilities?
  - Can the checklist be covered?
- Please provide feedback

References

- Details can be found in published and upcoming papers:

cocoa-00

  - A. Betzler, C. Gomez, I. Demirkol, J. Paradells, "Congestion Control in Reliable CoAP Communication", MSWIM'13, Barcelona, Spain, Nov. 2013.
  - A. Betzler, C. Gomez, I. Demirkol, M. Kovatsch, "Congestion Control for CoAP cloud services", 8th International Workshop on Service-Oriented Cyber-Physical Systems in Converging Networked Environments (SOCNE) 2014, Barcelona, Spain, Sept. 2014.

  - Two more papers under review

cocoa-02

  - Or you may contact the authors!