Contention Resolution in Multiple-access Channels: k-Selection in Radio Networks

Antonio Fernández Anta[2] Miguel A. Mosteiro[1,2]

[1]Department of Computer Science, Rutgers University

[2]LADyR, GSyC, Universidad Rey Juan Carlos

COCOON 2010

Shared Resource Contention

- Unique resource to be shared among many users
- All contenders must have access eventually
- Only one user at a time may have access

$\Longrightarrow$ Contention.

- Unknown number of contenders

$\Longrightarrow$ up to n.

- E.g.: k-Selection in Radio Networks:

  "unknown size-k subset of network nodes
  must access a unique shared channel of communication,
  each of them at least once."

- Q: k-Selection with unknown k in $O(k)$?

Shared Resource Contention

- Unique resource to be shared among many users
- All contenders must have access eventually
- Only one user at a time may have access

$\Longrightarrow$ Contention.

- Unknown number of contenders

$\Longrightarrow$ up to n.

- E.g.: k-Selection in Radio Networks:

  "unknown size-k subset of network nodes
  must access a unique shared channel of communication,
  each of them at least once."

- Q: k-Selection with unknown k in $O(k)$?

Shared Resource Contention

- Unique resource to be shared among many users
- All contenders must have access eventually
- Only one user at a time may have access

$\Longrightarrow$ Contention.

- Unknown number of contenders

$\Longrightarrow$ up to n.

- E.g.: k-Selection in Radio Networks:

  "unknown size-k subset of network nodes
  must access a unique shared channel of communication,
  each of them at least once."

- Q: k-Selection with unknown k in $O(k)$?

Shared Resource Contention

- Unique resource to be shared among many users
- All contenders must have access eventually
- Only one user at a time may have access

$\Longrightarrow$ Contention.

- Unknown number of contenders

$\Longrightarrow$ up to n.

- E.g.: k-Selection in Radio Networks:

  "unknown size-k subset of network nodes
  must access a unique shared channel of communication,
  each of them at least once."

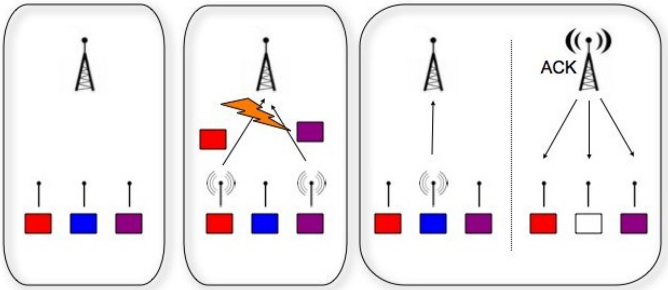
- Q: k-Selection with unknown k in $O(k)$?

Radio Network Model and Notation

- labeled stations called *nodes*.
- no information except own ID (unique, arbitrary) and n.
- time slotted in communication steps.
- nodes are potentially reachable in one comm step $\rightarrow$ *single-hop*.
- time complexity = communication steps (negligible computation cost).
- piece of information to deliver called *message*.
- node is *active* if holds a message to deliver.
- message assignment is external (called message *arrival*).
- *batched* message arrivals (*static* k-Selection).
- number of messages left to deliver called *density*.

Radio Network Model

- communication through radio broadcast on a shared channel.
- in one step:
  - no message transmitted → all nodes receive background noise.
  - more than one node transmits → all other nodes receive interference noise. (collision, messages garbled, etc.)
  - exactly one node transmits → all other nodes receive message and the sender receives an ack.
    (successful transmission, message delivered, single transmission, etc.)
- nodes can not distinguish between interference and background noise.



Related Work

Randomized

- With collision detection:
  - Willard'86:
    - Expected $\log\log k + o(\log\log k)$ for *Selection* with unknown n.
  - Martel'94:
    - k-Selection expected $O(k + \log n)$ with known n.
      Kowalski'05: can be improved to expected $O(k + \log\log n)$ using Willard's.
- Without collision detection:
  - Kushilevitz, Mansour'98:
    - For any given protocol, $\exists k$ s.t.
      expected $\Omega(\log n)$ to get even the first message delivered.

Related Work

Beyond Radio Networks

- Greenberg, Leiserson'89: *randomized routing of bounded number of messages in fat-trees* $\Rightarrow$ k-Selection if constant edge-capacities. $\Rightarrow$ logarithmic congestion parameter $\Rightarrow$ $O(k \text{ polylog } n)$.
- Gerèb-Graus, Tsantilas'92: *arbitrary h-relations realization* in $\Theta(h + \log n \log\log n)$ w.h.p. $\Rightarrow$ k-Selection if $h = k$. Needs h known.
- Bender, Farach-Colton, Kuszmaul, Leiserson'05: *Back-off strategies for contention resolution of batched arrivals of k packets on simple multiple access channels*
  log log-iterated back-off $\rightarrow \Theta(k \log\log k / \log\log\log k)$
  w.p. $\geq 1 - 1/k^{\Theta(1)}$, without knowledge of a bound on k.

Related Work

Deterministic

- *Tree* algorithms: $O(k \log(n/k))$ [H'78, MT'78, C'79]. Adaptive, with collision detection.
- Greenberg, Winograd'85: tree algorithms take $\Omega(k \log_k n)$.
- Greenberg, Komlòs'85: $\exists$ *oblivious* protocols $O(k \log(n/k))$ even without collision detection if k and n are known.
- Clementi, Monti, Silvestri'01: matching lower bound. Also holds for *adaptive* protocols if no collision detection.
- Kowalski'05: oblivious deterministic protocol $O(k \text{ polylog } n)$ without collision detection, using Indyk'02 explicit selectors.

Result

- Back-on/back-off Randomized k-Selection in one-hop Radio Network in $(e+1+\xi)k + O(\log^2(1/\varepsilon))$ with probability $\geq 1-\varepsilon$ unknown k ($\xi > 0$ arb. small constant).
  - Optimal (modulo a small constant factor < 4) if $\varepsilon \in \Omega(2^{-\sqrt{k}})$.
  - Given $\Omega(k \log\log k / \log\log\log k)$ [Bender et al.'05] for monotonic back-off, shows separation using back-on strategies.
  - Improves over loglog-iterated back-off $O(k \log\log k / \log\log\log k)$ [Bender et al.'05] by exploiting back-on and knowledge of n.
  - Error probability is parametric $\Rightarrow$ suitable to be used in multi-hop Radio Networks.

Protocol for node x

without constants

- Algorithm AT: (if $\delta > \log(1/\varepsilon)$ messages left)

  Concurrent Task 1:

  $t = \log(1/\varepsilon)$. (set up a step counter)

  $\hat{\delta} = \log(1/\varepsilon)$. (set up a density estimate)

  for each communication step

  transmit $\langle x, message \rangle$ with probability $1/\hat{\delta}$.

  $t = t - 1$.

  if $t \leq 0$

  $t = \log(1/\varepsilon)$. (new round)

  $\hat{\delta} = \hat{\delta} + \log(1/\varepsilon)$. (update estimate)

  Concurrent Task 2:

  upon receiving a message from other node

  $\hat{\delta} = \max\{\hat{\delta} - 1, \log(1/\varepsilon)\}$. (update estimate)

  $t = t + 1$. (stretch round)

  Concurrent Task 3:

  upon delivering message, stop.

Protocol for node x

without constants

- Algorithm AT: (if $\delta > \log(1/\varepsilon)$ messages left)

    Concurrent Task 1:

    $t = \log(1/\varepsilon)$. (set up a step counter)
    $\hat{\delta} = \log(1/\varepsilon)$. (set up a density estimate)
    for each communication step
        transmit $\langle x, message \rangle$ with probability $1/\hat{\delta}$.
        $t = t - 1$.
        if $t \leq 0$
            $t = \log(1/\varepsilon)$. (new round)
            $\hat{\delta} = \hat{\delta} + \log(1/\varepsilon)$. (update estimate)

    Concurrent Task 2:

    upon receiving a message from other node
        $\hat{\delta} = \max\{\hat{\delta} - 1, \log(1/\varepsilon)\}$. (update estimate)
        $t = t + 1$. (stretch round)

    Concurrent Task 3:

    upon delivering message, stop.

Protocol for node x

without constants

- Algorithm AT: (if $\delta > \log(1/\varepsilon)$ messages left)

  Concurrent Task 1:

  $t = \log(1/\varepsilon)$. (set up a step counter)

  $\hat{\delta} = \log(1/\varepsilon)$. (set up a density estimate)

  for each communication step

  transmit $\langle x, message \rangle$ with probability $1/\hat{\delta}$.

  $t = t - 1$.

  if $t \leq 0$

  $t = \log(1/\varepsilon)$. (new round)

  $\hat{\delta} = \hat{\delta} + \log(1/\varepsilon)$. (update estimate)

  Concurrent Task 2:

  upon receiving a message from other node

  $\hat{\delta} = \max\{\hat{\delta} - 1, \log(1/\varepsilon)\}$. (update estimate)

  $t = t + 1$. (stretch round)

  Concurrent Task 3:

  upon delivering message, stop.

Protocol for node x

without constants

- Algorithm BT: (if $\delta \leq \log(1/\varepsilon)$ messages left)

  for each communication step

  transmit $\langle x, message \rangle$ with probability $1/\log(1/\varepsilon)$.

Protocol for node x

without constants

But we do not know δ $\rightarrow$ interleave both:

Concurrent Task 1:
  $t = \log(1/\varepsilon)$, $\hat{\delta} = \log(1/\varepsilon)$.
  for each communication step
    if step is even (Algorithm BT)
      transmit $\langle x, message\rangle$ with probability $1/\log(1/\varepsilon)$.
    if step is odd (Algorithm AT)
      transmit $\langle x, message\rangle$ with probability $1/\hat{\delta}$.
      $t = t - 1$.
      if $t \leq 0$
        $t = \log(1/\varepsilon)$, $\hat{\delta} = \hat{\delta} + \log(1/\varepsilon)$.
Concurrent Task 2:
  upon receiving a message from other node
    $\hat{\delta} = \max\{\hat{\delta} - 1, \log(1/\varepsilon)\}$.
    $t = t + 1$.
Concurrent Task 3:
  upon delivering message, stop.

Analysis

Time Efficiency

- Failure steps:
  - Algorithm AT:
    - At most $k/\log(1/\varepsilon)$ rounds until the estimate reaches k.
    - Each round includes $\log(1/\varepsilon)$ failure steps.
  - Algorithm BT:
    - After $\leq \log(1/\varepsilon)$ messages are left, in $O(\log^2(1/\varepsilon))$ steps w.p. $1-\varepsilon$ BT delivers all.
- Success steps:
  - One step per message delivered $\Rightarrow k$ overall.
- Overall:
  - $(e+1+\xi)k + O(\log^2(1/\varepsilon))$
    (there are some constants...)

Analysis

Time Efficiency

- Failure steps:
  - Algorithm AT:
    - At most $k/\log(1/\varepsilon)$ rounds until the estimate reaches k.
    - Each round includes $\log(1/\varepsilon)$ failure steps.
  - Algorithm BT:
    - After $\leq \log(1/\varepsilon)$ messages are left, in $O(\log^2(1/\varepsilon))$ steps w.p. $1-\varepsilon$ BT delivers all.
- Success steps:
  - One step per message delivered $\Rightarrow k$ overall.
- Overall:
  - $(e+1+\xi)k + O(\log^2(1/\varepsilon))$
    (there are some constants...)

Analysis

Time Efficiency

- Failure steps:
  - Algorithm AT:
    - At most $k/\log(1/\varepsilon)$ rounds until the estimate reaches k.
    - Each round includes $\log(1/\varepsilon)$ failure steps.
  - Algorithm BT:
    - After $\leq \log(1/\varepsilon)$ messages are left, in $O(\log^2(1/\varepsilon))$ steps w.p. $1-\varepsilon$ BT delivers all.
- Success steps:
  - One step per message delivered $\Rightarrow k$ overall.
- Overall:
  - $(e+1+\xi)k + O(\log^2(1/\varepsilon))$
    (there are some constants...)

Analysis

Time Efficiency

- Failure steps:
  - Algorithm AT:
    - At most $k/\log(1/\varepsilon)$ rounds until the estimate reaches k.
    - Each round includes $\log(1/\varepsilon)$ failure steps.
  - Algorithm BT:
    - After $\leq \log(1/\varepsilon)$ messages are left, in $O(\log^2(1/\varepsilon))$ steps w.p. $1-\varepsilon$ BT delivers all.
- Success steps:
  - One step per message delivered $\Rightarrow k$ overall.
- Overall:
  - $(e+1+\xi)k + O(\log^2(1/\varepsilon))$
    (there are some constants...)

Analysis

Correctness sketch

Lemma

Until the estimate $\hat{\delta}$ is some "constant close" to the number of messages left δ, within a round, the probability of passing a message is not positively correlated with time.

Lemma

If $\hat{\delta}$ is "logarithmically close" to δ at the begining of a round, until the number of messages delivered in this round is logarithmic, the probability of delivering a message is at least constant.

$\Longrightarrow$ we can bound from below logarithmically the number of messages delivered in each round where the estimate is "close" to the number of messages left using Chernoff.

Analysis

Correctness sketch

Lemma

Until the estimate $\hat{\delta}$ is some "constant close" to the number of messages left δ, within a round, the probability of passing a message is not positively correlated with time.

Lemma

If $\hat{\delta}$ is "logarithmically close" to δ at the begining of a round, until the number of messages delivered in this round is logarithmic, the probability of delivering a message is at least constant.

$\Longrightarrow$ we can bound from below logarithmically the number of messages delivered in each round where the estimate is "close" to the number of messages left using Chernoff.

Analysis

Correctness sketch

Lemma

If δ is more than some logarithmic number M, after running the AT-algorithm for $(e+1+\xi)k$ steps, where $\xi > 0$ is any constant arbitrarily close to 0, $\delta \leq M$ with probability at least $1-\varepsilon$.

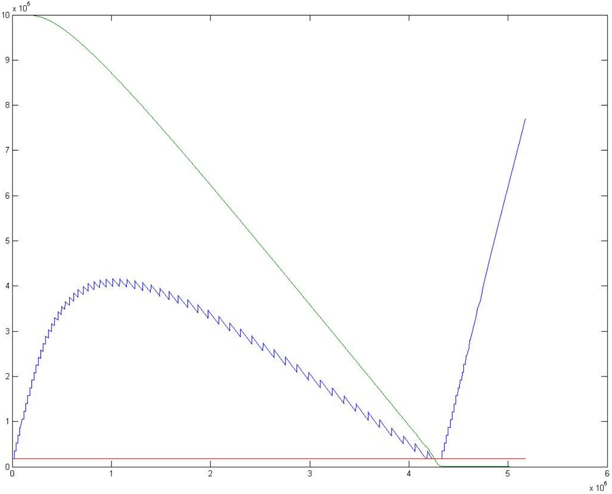
Proof.

- Using Chernoff, the probability of reducing δ logarithmically in each round after $\hat{\delta}$ is "logarithmically close" is at least $1-\varepsilon/(k+\varepsilon)$.
- Using an inductive argument over rounds and conditional probabilities, $\delta \leq M$ in one pass with probability at least $1-\varepsilon$.

□

Analysis

Illustration of estimate progress



Analysis

Correctness sketch

Together with Algorithm BT...

Theorem

For any one-hop Radio Network, under the model detailed, the protocol described solves the k-selection problem within $(e+1+\xi)k + O(\log^2(1/\varepsilon))$ communication steps, where $\xi > 0$ is any constant arbitrarily close to 0, with probability at least $1-\varepsilon$.

Application

Farach-Colton, Fernández Anta, Mosteiro
Optimal Memory-Aware Sensor Network Gossiping

Phases of the algorithm:

1. Partition nodes in *masters* and *slaves*
   $\Rightarrow$ MIS $\rightarrow O(\log^2 n)$
2. Every master reserves blocks of time steps for local use
   $\Rightarrow$ Coloring $\rightarrow O(\log n)$
3. Every master maintains set of messages received
   $\Rightarrow$ back-on/back-off $\rightarrow O(\Delta + \log^2 n)$
4. Every master disseminates local set
   $\Rightarrow$ flooding among masters $\rightarrow O(D)$

Overall:

$$O(\log^2 n + \log n + \Delta + \log^2 n \log \Delta + D) \in O(\Delta + D) \text{ w.h.p.}$$

Future Work

1. Remove the knowledge of n.
2. Generalize the system model:
   - Continuous message arrival.
   - Arbitrary wake-up.
3. Evaluation of the algorithm:
   - Study its practicality by simulation.
   - Compare it with other algorithms (e.g., Bender et al).
   - Improve its constants.

Thank you