

Unbounded Contention Resolution in Multiple-access Channels

Antonio Fernández Anta
Institute IMDEA Networks



Miguel A. Mosteiro
Rutgers U. and U. Rey Juan Carlos

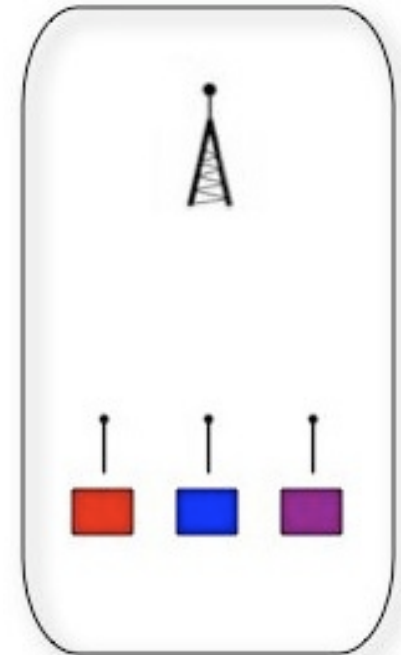
Jorge Ramón Muñoz
U. Rey Juan Carlos

Shared Resource Contention

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

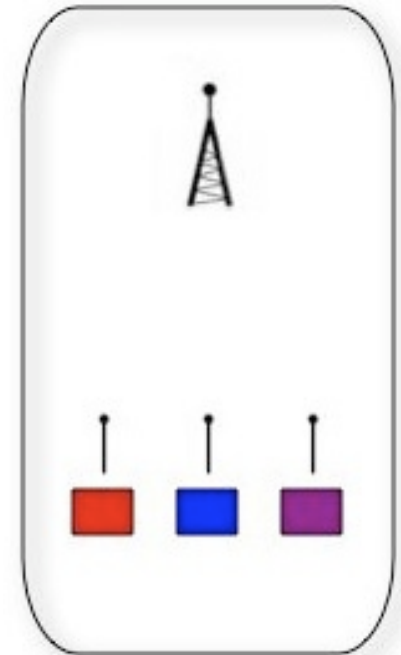
Shared Resource Contention

- Unique resource to be shared among users
 - All contenders must have access eventually
 - Only one user at a time may have access
 - This leads to **contention**
- **k -Selection problem** in Radio Networks:
 - k out of n network nodes have one message to deliver to a base station (BS)
 - Shared communication channel
 - Each node eventually delivers to the BS



Shared Resource Contention

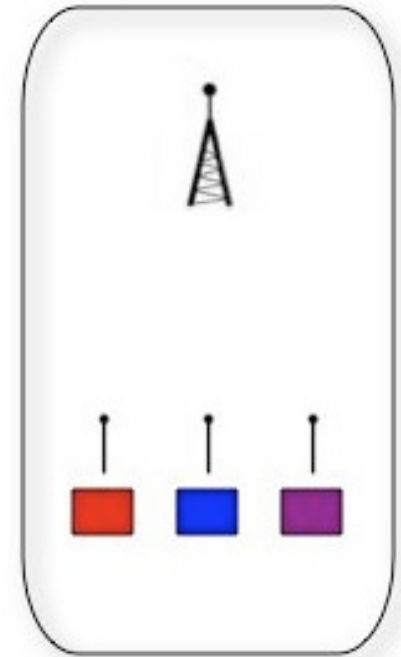
- Unique resource to be shared among users
 - All contenders must have access eventually
 - Only one user at a time may have access
 - This leads to **contention**
- **k -Selection problem** in Radio Networks:
 - k out of n network nodes have one message to deliver to a base station (BS)
 - Shared communication channel
 - Each node eventually delivers to the BS



Q: Efficient k -selection with n and k unknown?

Shared Resource Contention

- Unique resource to be shared among users
 - All contenders must have access eventually
 - Only one user at a time may have access
 - This leads to **contention**
- **k -Selection problem** in Radio Networks:
 - k out of n network nodes have one message to deliver to a base station (BS)
 - Shared communication channel
 - Each node eventually delivers to the BS



Q: Efficient k -selection with n and k unknown?

We concentrate on randomized protocols

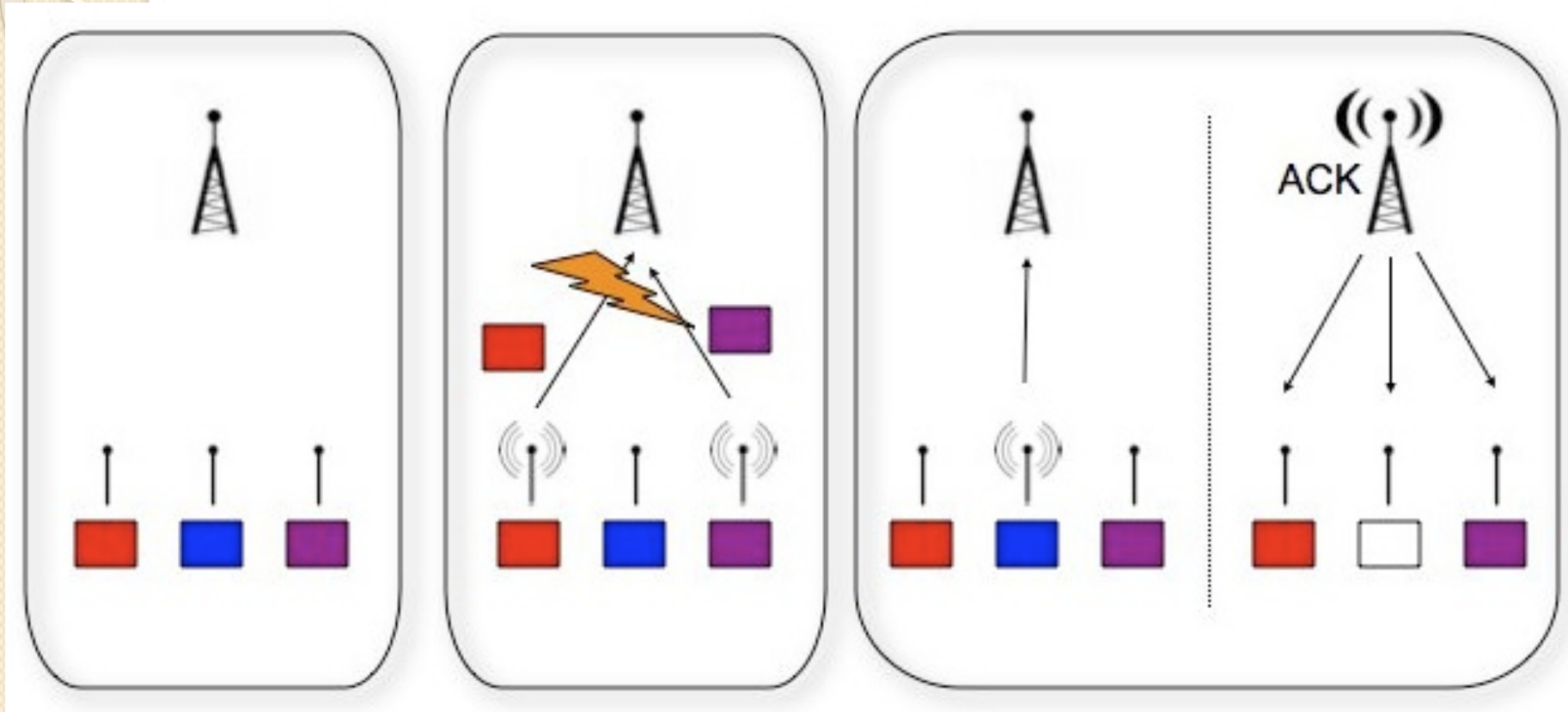
Radio Network Model

- All nodes are reachable in one comm step:
single-hop
- No network information (not even the node ID)
- **Time slotted** in communication steps
- Time complexity = communication steps to complete (no computation cost)
- **Batched message arrivals** (static k -Selection): all messages arrive at the same time

Radio Network Model

- Broadcast on a shared channel:
 - No node transmits → background noise
 - More than one node transmits → collision
 - Exactly one node transmits → BS receives the message and broadcasts an ACK (successful transmission, message delivered)
- Nodes can not distinguish between collision and background noise: no collision detection.

Radio Network Model



Background noise

Collision

Successful transmission

Related Work

Deterministic protocols:

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

Related Work

Randomized protocols:

- With collision detection:
 - Willard'86: Expected $\log \log k + o(\log \log k)$ for delivering the first message, with unknown n
 - Martel'94: k -selection expected $O(k + \log n)$ with known n
 - Kowalski'05: can be improved using Willard's protocol to expected $O(k + \log \log n)$
- 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 :

- Gerèb-Graus, Tsantilas'92: arbitrary k -relations realization in $\Theta(k + \log n \log \log n)$ w.h.p. (known k)
- Greenberg, Leiserson'89: routing of messages in fat-trees

Farach-Colton, Mosteiro'07: sensor network gossiping

- Sawtooth technique embedded
- Known n , asymptotic analysis.

Related Work

Randomized protocols without collision detection:

- Bender, Farach-Colton, He, Kuszmaul, Leiserson'05: back-off strategies
 - Loglog-iterated Back-off:
 - $\Theta(k \log \log k / \log \log \log k)$ w.p. $\geq 1 - 1/k^{\Theta(1)}$
 - Unknown k and n
 - Linear sawtooth technique described, no analysis
- Fernández Anta, Mosteiro'10: back-on/back-off
 - Log-fails Adaptive: $8k + O(\log^2 (1/\varepsilon))$, w.p. $\geq 1 - 2\varepsilon$
 - Known n (ε depends on n)



Related Work

Randomized protocols without collision detection:

- Bender, Farach-Colton, He, Kuszmaul, Leiserson'05: back-off strategies
 - Loglog-iterated Back-off:
 - $\Theta(k \log \log k / \log \log \log k)$ w.p. $\geq 1 - 1/k^{\Theta(1)}$ Not linear on k
 - Unknown k and n
 - Linear sawtooth technique described, no analysis
- Fernández Anta, Mosteiro'10: back-on/back-off
 - Log-fails Adaptive: $8k + O(\log^2(1/\varepsilon))$, w.p. $\geq 1 - 2\varepsilon$
 - Known n (ε depends on n)

Related Work

Randomized protocols without collision detection:

- Bender, Farach-Colton, He, Kuszmaul, Leiserson'05: back-off strategies
 - Loglog-iterated Back-off:
 - $\Theta(k \log \log k / \log \log \log k)$ w.p. $\geq 1 - 1/k^{\Theta(1)}$ 
 - Unknown k and n
 - Linear sawtooth technique described, no analysis
- Fernández Anta, Mosteiro'10: back-on/back-off
 - Log-fails Adaptive: $8k + O(\log^2(1/\epsilon))$, w.p. $\geq 1 - 2\epsilon$
 - Known n (ϵ depends on n) 

Not linear on k

What if n is not known?

Results

- Randomized k -selection protocols:
 - One-fail Adaptive:
$$\approx 2(e + 1)k + O(\log^2 k), \quad \text{w.p.} \geq 1 - 2/(1+k)$$
 - Exponential Back-on/Back-off:
$$\approx 4(e + 1)k, \quad \text{w.p.} \geq 1 - 1/k^{\Theta(1)}$$
- Time-optimal & unknown k and n
 - Improve Log-fails Adaptive removing knowledge of n
 - Sawtooth analyzed down to constants

One-fail Adaptive

- State at each node:
 - An estimate (lower bound) α of the number of active nodes (density δ)
 - Number of successful transmissions σ (lower bound on k)
- In odd steps (Algorithm AT)
 - Nodes transmit with probability $1/\alpha$
- In even steps (Algorithm BT)
 - Nodes transmit with probability $\approx 1/\log \sigma$

One-fail Adaptive

- Intuition:
 - When there are $\delta > \log k$ active nodes and α gets close to δ , AT steps are “good”
 - When there are $\delta \leq \log k$ active nodes, $\sigma \approx k$, and BT steps are “good”
- It is important to keep α below δ as long as active nodes $\delta > \log k$

One-fail Adaptive

Protocol for node x (without constants):

Concurrent Task 1:

$$\sigma = 0, \alpha = e$$

for each communication step

if step is even

(Algorithm BT)

transmit $\langle \text{message} \rangle$ with probability $1/\log \sigma$

if step is odd

(Algorithm AT)

transmit $\langle \text{message} \rangle$ with probability $1/\alpha$

$$\alpha = \alpha + 1$$

Concurrent Task 2:

upon receiving an ACK from BS

if x did transmit then stop

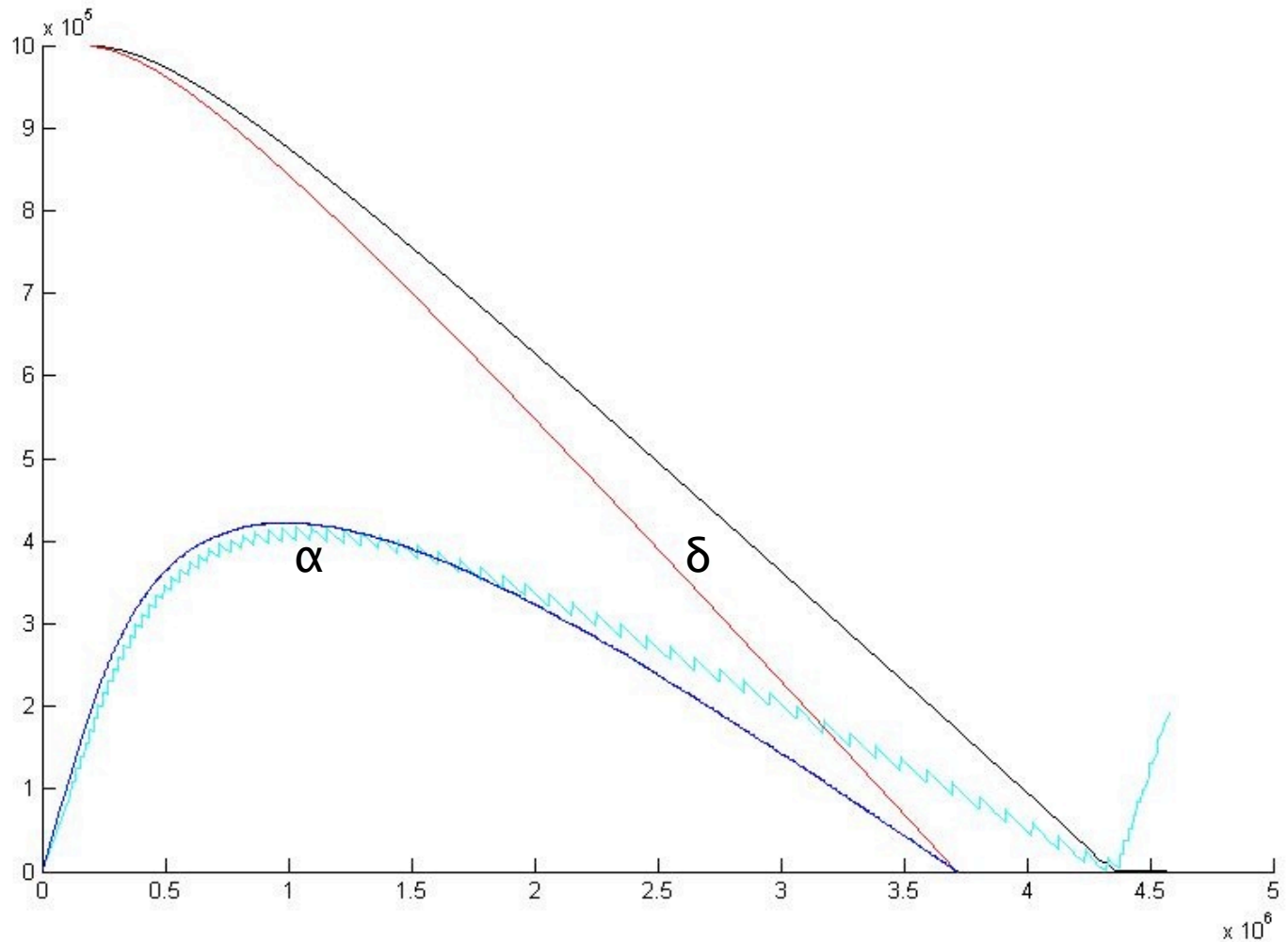
(successful transmission)

$$\sigma = \sigma + 1$$

$$\alpha = \max \{ \alpha - e, e \}$$

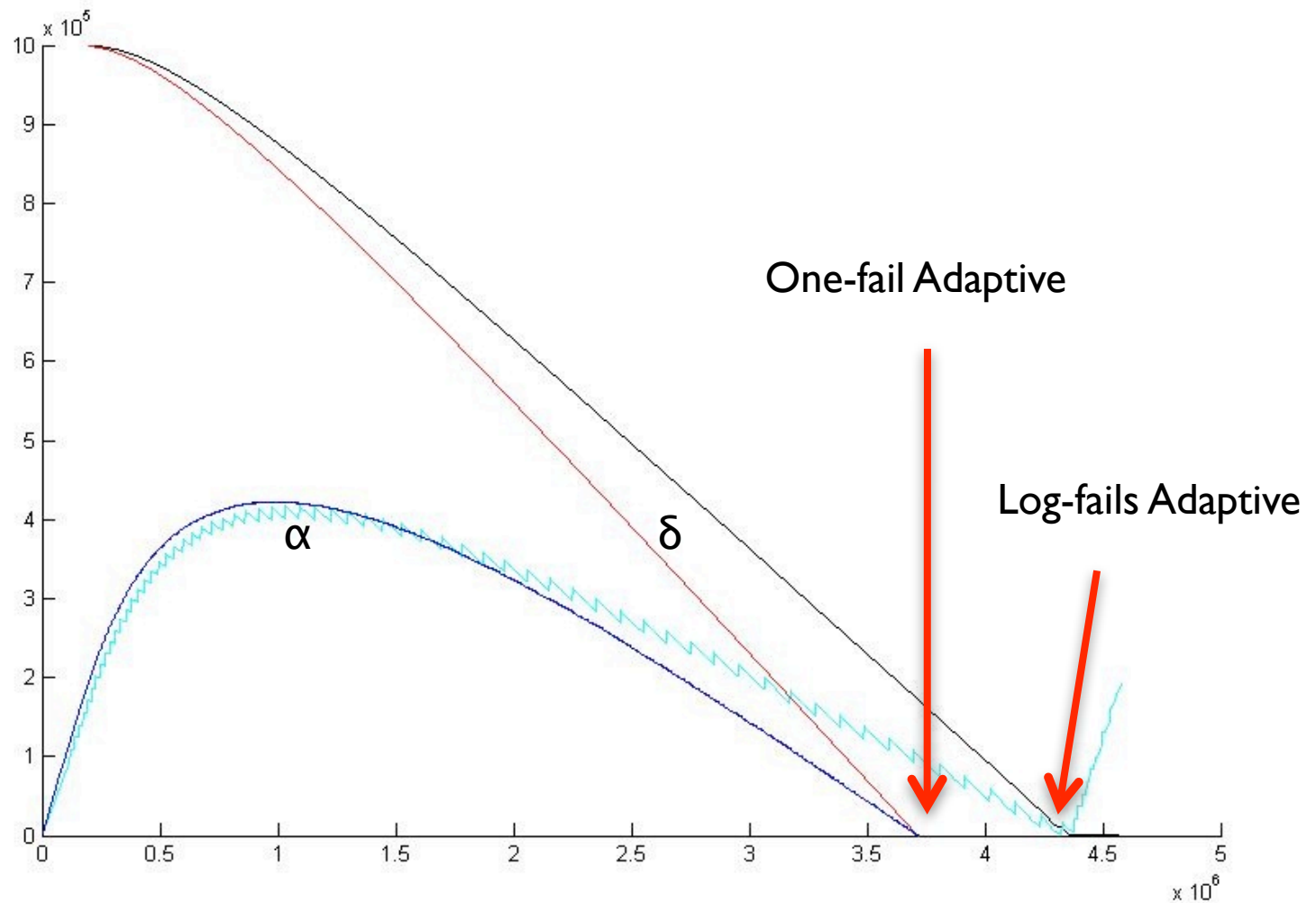
One-fail Adaptive

Estimate evolution (Algorithm AT)



One-fail Adaptive

Estimate evolution (Algorithm AT)



One-fail Adaptive

Correctness:

- Algorithm AT:
 - We divide the time into rounds of $\approx \log k$ steps
 - Concentration bounds show that $\geq \log k$ messages/round are delivered if $\alpha \approx \delta$
 - Then, estimate α never exceeds the density δ
- Algorithm BT:
 - When density $\delta \leq \log k$, then $\sigma = \Theta(k)$ and rest of messages delivered in $O(\log^2 k)$ steps w.p. $1 - 1/k$

One-fail Adaptive

Time performance:

- Algorithm AT:
 - Initially, density – estimate: $\delta - \alpha < k$
 - Difference $\delta - \alpha$ increased with each message delivered by at most e
 - Difference decreases by 1 otherwise
 - But estimate α always $<$ density δ
 - Hence at most $(e + 1)k$ AT steps
- Algorithm BT: $O(\log^2 k)$ steps
- Overall:
 - $2(e + 1)k + O(\log^2 k) \approx 7.4 k + O(\log^2 k)$

Exponential Back-on/Back-off

Window size adjustment:

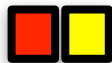
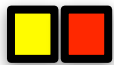
for $i = 1, 2, \dots$

$$w = 2^i$$

while $w \geq 1$

transmit in a uniformly chosen step in next w steps

$$w = w (1 - 1/e)$$



Exponential Back-on/Back-off

- Correctness:
 - Bins and balls argument to show at least a constant fraction of deliveries in each sub-round after $k < w \leq 2k$
 - The process completes in the round $w = 4k$

Time performance:

- Telescoping the number of steps up to the first round when $w = 4k$ yields

$$4(e + 1)k \approx 14.9 k$$



Simulations

Simulations

k	10	10^2	10^3	10^4	10^5	10^6	10^7	Analysis
LOG-FAILS ADAPTIVE $\xi_t = 1/2$	46.4	1292.4	181.9	26.6	9.4	8.0	7.8	7.8
LOG-FAILS ADAPTIVE $\xi_t = 1/10$	26.3	3289.2	593.8	50.3	11.5	4.5	4.4	4.4
ONE-FAIL ADAPTIVE	4.0	6.9	7.4	7.4	7.4	7.4	7.4	7.4
EXP BACK-ON/BACK-OFF	4.0	5.5	5.2	7.2	6.6	5.6	7.9	14.9
LOGLOG-ITERATED BACK-OFF	5.6	8.6	9.6	9.2	10.5	10.5	10.1	$\Theta\left(\frac{\log \log k}{\log \log \log k}\right)$

Ratio steps/nodes as a function of the number of nodes k .

Simulations

Bad for small k



k	10	10^2	10^3	10^4	10^5	10^6	10^7	Analysis
LOG-FAILS ADAPTIVE $\xi_t = 1/2$	46.4	1292.4	181.9	26.6	9.4	8.0	7.8	7.8
LOG-FAILS ADAPTIVE $\xi_t = 1/10$	26.3	3289.2	593.8	50.3	11.5	4.5	4.4	4.4
ONE-FAIL ADAPTIVE	4.0	6.9	7.4	7.4	7.4	7.4	7.4	7.4
EXP BACK-ON/BACK-OFF	4.0	5.5	5.2	7.2	6.6	5.6	7.9	14.9
LOGLOG-ITERATED BACK-OFF	5.6	8.6	9.6	9.2	10.5	10.5	10.1	$\Theta\left(\frac{\log \log k}{\log \log \log k}\right)$

Ratio steps/nodes as a function of the number of nodes k .

Simulations

good even for small k



k	10	10^2	10^3	10^4	10^5	10^6	10^7	Analysis
LOG-FAILS ADAPTIVE $\xi_t = 1/2$	46.4	1292.4	181.9	59.6	9.4	8.0	7.8	7.8
LOG-FAILS ADAPTIVE $\xi_t = 1/10$	26.3	3289.2	593.8	50.3	11.5	4.5	4.4	4.4
ONE-FAIL ADAPTIVE	4.0	6.9	7.4	7.4	7.4	7.4	7.4	7.4
EXP BACK-ON/BACK-OFF	4.0	5.5	5.2	7.2	6.6	5.6	7.9	14.9
LOGLOG-ITERATED BACK-OFF	5.6	8.6	9.6	9.2	10.5	10.5	10.1	$\Theta\left(\frac{\log \log k}{\log \log \log k}\right)$

Ratio steps/nodes as a function of the number of nodes k .

Simulations

k	10	10^2	10^3	10^4	10^5	10^6	10^7	Analysis
LOG-FAILS ADAPTIVE $\xi_t = 1/2$	46.4	1292.4	181.9	26.6	9.4	8.0	7.8	7.8
LOG-FAILS ADAPTIVE $\xi_t = 1/10$	26.3	3289.2	593.8	50.3	11.5	4.5	4.4	4.4
ONE-FAIL ADAPTIVE	4.0	6.9	7.4	7.4	7.4	7.4	7.4	7.4
EXP BACK-ON/BACK-OFF	4.0	5.5	5.2	7.2	6.6	5.6	7.9	14.9
LOGLOG-ITERATED BACK-OFF	5.6	8.6	9.6	9.2	10.5	10.5	10.1	$\Theta\left(\frac{\log \log k}{\log \log \log k}\right)$

Ratio steps/nodes as a function of the number of nodes k .

Not far from the optimal ratio e

Simulations

k	10	10^2	10^3	10^4	10^5	10^6	10^7	Analysis
LOG-FAILS ADAPTIVE $\xi_t = 1/2$	46.4	1292.4	181.9	26.6	9.4	8.0	7.8	7.8
LOG-FAILS ADAPTIVE $\xi_t = 1/10$	26.3	3289.2	593.8	50.3	11.5	4.5	4.4	4.4
ONE-FAIL ADAPTIVE	4.0	6.9	7.4	7.4	7.4	7.4	7.4	7.4
EXP BACK-ON/BACK-OFF	4.0	5.5	5.2	7.2	6.6	5.6	7.9	14.9
LOGLOG-ITERATED BACK-OFF	5.6	8.6	9.6	9.2	10.5	10.5	10.1	$\Theta\left(\frac{\log \log k}{\log \log \log k}\right)$

Ratio steps/nodes as a function of the number of nodes k .

Conclusions

- Static k -selection solvable in almost optimal time
- Even without knowledge of k nor n
- Probability w.r.t. k , not with n
- n is not used!!
- Work in progress:
 - Packet arrivals not simultaneous
 - Continuous packet arrival
 - 802.11-friendly algorithms



Thank you!!

One-fail Adaptive

Protocol for node x (without constants)

Concurrent Task 1:

$\sigma = 0, \hat{\kappa} = 4$. (msg-received counter, density estimate)

for each communication step

if step is even (Algorithm BT)

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

if step is odd (Algorithm AT)

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

$\hat{\kappa} = \hat{\kappa} + 1$. (new estimate)

Concurrent Task 2:

upon receiving an ACK from BS

$\sigma = \sigma + 1$. (update counter)

if step is even (Algorithm BT)

$\hat{\kappa} = \max\{\hat{\kappa} - 3, 4\}$. (new estimate)

if step is odd (Algorithm AT)

$\hat{\kappa} = \max\{\hat{\kappa} - 4, 4\}$. (new estimate)

Concurrent Task 3:

upon delivering message, stop.

Exponential Back-on/Back-off

Window size adjustment

for $i = \{1, 2, \dots\}$

$w = 2^i$

while $w \geq 1$

choose uniformly a step within the next w steps

$w = w(1 - 1/e)$