# Initialization for Ad Hoc Radio Networks with Carrier Sensing and Collision Detection

Jacek Cichoń Mirosław Kutyłowski Marcin Zawada

Institute of Mathematics and Computer Science
Wrocław University of Technology
Poland

5th International Conference on AD-HOC Networks & Wireless 2006

# Outline

# Problem

## Initialization problem

The initialization is a task of assigning to the $n$ stations of the radio network distinct $ID$ numbers in the range 1 to $n$.

# Outline

## Carrier sensing

1. we are able to sense the channel as either **busy** or **idle**

## Carrier sensing

1. we are able to sense the channel as either **busy** or **idle**
2. signal has a propagation delay

## Carrier sensing

1. we are able to sense the channel as either **busy** or **idle**
2. signal has a propagation delay
3. IEEE 802.11

# Outline

# Introduction
## Previous randomized algorithms

- Nakano-Olariu (2000) : without carrier sensing
  - Known number of stations
    Time: $2.718 \cdot n + O(\sqrt{n \log n})$
  - Unknown number of stations
    Time: $3.333 \cdot n + O(\sqrt{n \log n})$
- Cai-Lu-Wang (2003) : with carrier sensing
  - Known number of stations
  - Unknown number of stations

Probability at least $1 - \frac{1}{n}$. Time complexity of Cai-Lu-Wang algorithms are better than of Nakano-Olariu ones and will be discussed later.

# Outline

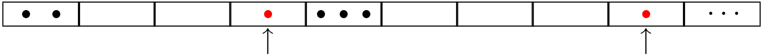
# Nakano - Olariu algorithm

## Known number of stations



### Sketch of algorithm

**for** $m := n$ **downto** 1 **do**
    **repeat**
        each station transmit with probability $\frac{1}{m}$
    **until** channel is SINGLE
    single station sets $id := n - m + 1$ and leaves



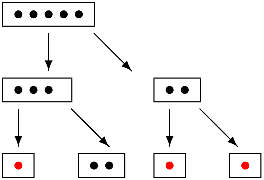
We should play $2.718 \cdot n + O(\sqrt{n \log n})$ times if we want each station to transmit in some slot.

# Nakano - Olariu algorithm

## Unknown number of stations

### Description of the algorithm

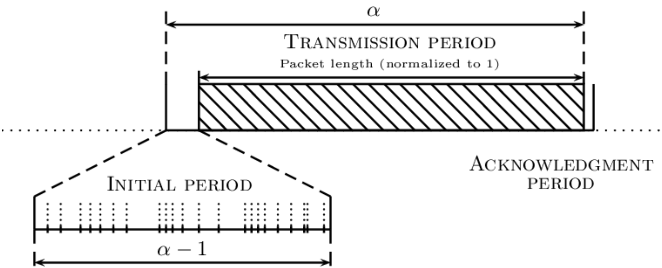
There are $n$ stations. They are divided into groups. If only one station is in the group we called it a **winner**. If not, then each station from the group flips a coin with probability $\frac{1}{2}$ and according to the result goes into a subgroup.



We should play $3.333 \cdot n + O(\sqrt{n \log n})$ times if we want each station to win in some slot (with probability at least $1 - \frac{1}{n}$).

# Outline

## The slot structure



# Outline

# Known number of stations

Fix probability $p$. The following algorithm is executed in each slot.

# Known number of stations

Fix probability $p$. The following algorithm is executed in each slot.

## Basic idea

1. each station with probability $p$ chooses a random time $X_i$ at which point it will check the channel.

# Known number of stations





Fix probability $p$. The following algorithm is executed in each slot.

**Basic idea**

1. each station with probability $p$ chooses a random time $X_i$ at which point it will check the channel.
2. if the channel is **idle** then the station starts transmitting a short signal

# Known number of stations

Fix probability $p$. The following algorithm is executed in each slot.

## Basic idea

1. each station with probability $p$ chooses a random time $X_i$ at which point it will check the channel.
2. if the channel is **idle** then the station starts transmitting a short signal
3. if there is no collision then the station transmits workload packet otherwise it stops transmission in the slot

# Known number of stations

Fix probability $p$. The following algorithm is executed in each slot.

## Basic idea

1. each station with probability $p$ chooses a random time $X_i$ at which point it will check the channel.
2. if the channel is **idle** then the station starts transmitting a short signal
3. if there is no collision then the station transmits workload packet otherwise it stops transmission in the slot
4. go to the next slot with remaining stations

# Known number of stations



Fix probability $p$. The following algorithm is executed in each slot.

## Basic idea

1. each station with probability $p$ chooses a random time $X_i$ at which point it will check the channel.
2. if the channel is **idle** then the station starts transmitting a short signal
3. if there is no collision then the station transmits workload packet otherwise it stops transmission in the slot
4. go to the next slot with remaining stations

What is the optimal probability $p^*$?

# Outline

# Known number of stations

## Good configurations

The probability of successful transmission is equal

$$\Pr[(\exists\, 1 \leqslant i \leqslant n)\, (X_{(i)} - X_{(i-1)} > \delta, X_{(i+1)} - X_{(i)} > \delta)]$$

where $X_i$ (for $i \in [1, n]$) is a random time selected by station and $\delta$ is the propagation delay.

# Known number of stations

## Good configurations

The probability of successful transmission is equal

$$\Pr[(\exists 1 \leqslant i \leqslant n)\,(X_{(i)} - X_{(i-1)} > \delta, X_{(i+1)} - X_{(i)} > \delta)]$$

where $X_i$ (for $i \in [1, n]$) is a random time selected by station and $\delta$ is the propagation delay.

How to calculate the probability?

# Known number of stations
## Combinatorial classes



We consider a discretization of this problem:

# Known number of stations

## Combinatorial classes

We consider a discretization of this problem:

We use the technology of combinatorial classes:

$$S(\circ) \times (\bullet \times S_{<D}(\circ))^a \times (\bullet \times S_{\geq D}(\circ))^2 \times (\bullet \times S(\circ))^{n-2-a}$$

Its generating function is $F_a(z) = \frac{(1-z^D)^a z^{2D} z^n}{(1-z)^{n+1}}$. Binomial identities, Stirling numbers, going back to continuous model:

We consider a discretization of this problem:

$$\circ\circ\circ\circ\circ\circ\circ\circ\circ\circ\circ\bullet\circ\circ\bullet\circ\circ\circ\circ\circ\circ\bullet\circ\circ\circ\bullet\circ\circ\bullet\circ\bullet\circ\circ\circ\circ\circ\circ\circ\circ\circ\bullet\circ\circ\circ\circ\circ\circ\circ\circ\circ$$

We use the technology of combinatorial classes:

$$S(\circ) \times (\bullet \times S_{<D}(\circ))^a \times (\bullet \times S_{\geq D}(\circ))^2 \times (\bullet \times S(\circ))^{n-2-a}$$

Its generating function is $F_a(z) = \frac{(1-z^D)^a z^{2D} z^n}{(1-z)^{n+1}}$. Binomial identities, Stirling numbers, going back to continuous model:

**Theorem**

$P[success] \approx 2(1-\delta)^n - (1-2\delta)^n$

# Known number of stations

## Adding flexibility

Now: each station transmits in a slot with probability $p = \frac{a}{n}$. Then

$$P[success] \geq 2(1 - \frac{\delta a}{n})^n - (1 - \frac{2\delta a}{n})^n - (1 - \frac{a}{n})^n.$$

Now: each station transmits in a slot with probability $p = \frac{a}{n}$. Then

$$P[success] \geq 2(1 - \frac{\delta a}{n})^n - (1 - \frac{2\delta a}{n})^n - (1 - \frac{a}{n})^n .$$

Using Chernoff bound we get

**Theorem**

*If $a \approx \ln(\frac{1}{2\delta^2}) - \ln\ln(\frac{1}{2\delta^2})$ then after $\frac{1}{1-\delta^2} n + O(\sqrt{n \ln n})$ slots each station transmit with probability at least $1 - \frac{1}{n}$.*

# Outline

# Known number of stations

## Comparison

### Comparison

CLW: Cai-Lu-Wang algorithm
CKZ: Cichon-Kutylowski-Zawada algorithm

| $\lambda$ | CLW (2003) | CKZ (2006) |
|---|---|---|
| 0.00001 | $1.0177 \cdot n$ | $1.00088 \cdot n$ |
| 0.0001 | $1.0500 \cdot n$ | $1.00400 \cdot n$ |
| 0.001 | $1.1500 \cdot n$ | $1.01900 \cdot n$ |

Time complexity of the old solution of Nakamo and Olariu is $2.781 \cdot n$

# Outline

# Unknown number of stations
## Sketch of algorithm

Fix probability $p$.

**Basic idea**

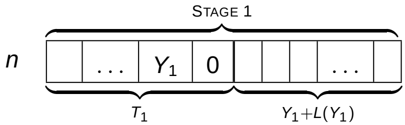
1. all stations flip a coin with probability of success $p$
2. we repeat round (1) until all stations fail
3. stations from **last but one** round we call **winners**, they use the strategy from our previous algorithm
4. go back to (1) with remaining stations

What is the optimal probability $p^*$?

# Sketch of the algorithm

## What we need to calculate



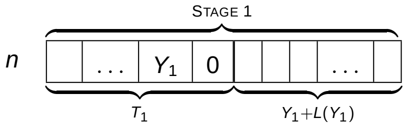



### Main steps

1. the number of winners $Y_1(n)$

# Sketch of the algorithm

## What we need to calculate

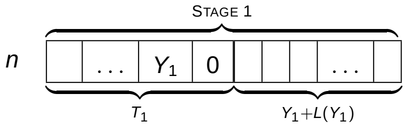



### Main steps

1. the number of winners $Y_1(n)$
2. the number of rounds $T_1(n)$

# Sketch of the algorithm

## What we need to calculate

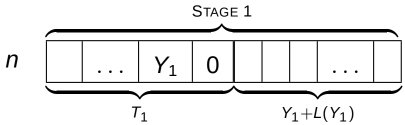



### Main steps

1. the number of winners $Y_1(n)$
2. the number of rounds $T_1(n)$
3. the number of collisions in second part of the stage $L_1(n)$

# Sketch of the algorithm

## What we need to calculate



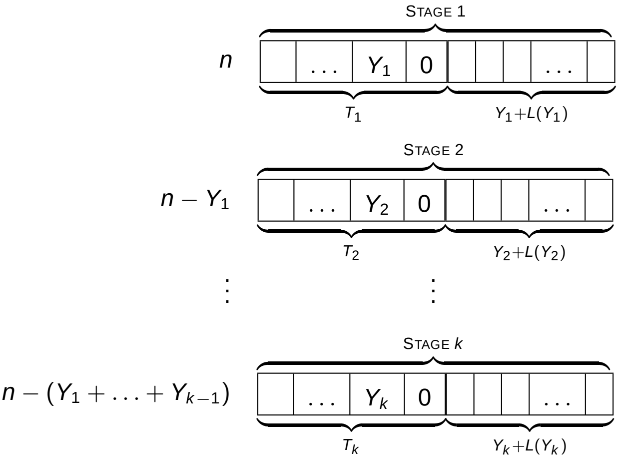

### Main steps

1. the number of winners $Y_1(n)$
2. the number of rounds $T_1(n)$
3. the number of collisions in second part of the stage $L_1(n)$
4. the total number of additional slots $H_1(n) = T_1(n) + L(Y_1(n))$

# Sketch of the algorithm
## What we need to calculate







# Outline

Initialization for Ad Hoc Radio Networks

Cichoń, Kutyłowski, Zawada

Known number of stations
Algorithm
Analysis
Comparison

Unknown number of stations
Algorithm
Analysis
Comparison

**Theorem**

*Let $Y(n)$ be random variable denoting the number of winners, where n is the number of stations. Then*

$$E[Y(n)] = \frac{n(1-p)}{p\ln(1/p)}\left(\frac{1}{n} + 2\sum_{k=1}^{\infty}\Re[B(n, 1+\frac{2k\pi i}{\ln(p)})]\right)$$

*where*

$$B(n,z) = \frac{\Gamma(n)\Gamma(z)}{\Gamma(n+z)}.$$

# Unknown number of stations

## Number of rounds

**Theorem**

*Let $T(n)$ be a random variable denoting the number of rounds such that the number of winners becomes 0, when we start with n stations. Then*

$$\mathbf{E}[T(n)] = \frac{1}{2} + \frac{H_n}{\log(1/p)} + \frac{2}{\log(1/p)} \sum_{k=1}^{\infty} \Re\left[\mathrm{B}\left(n+1, \frac{2k\pi i}{\log(p)}\right)\right]$$

*where $H_n = \sum_{k=1}^{n} \frac{1}{k}$ is the n-th harmonic number.*

# Unknown number of stations

## Number of wasted slots

**Theorem**

*Let $L(Y(n))$ be the number of additional slots in the second part of the stage. Then $E[L(Y(n))]\frac{\ln(1/p)}{n(1-p)}$ equals*

$$\frac{1}{n(p-\delta)} - \frac{1}{pn} + \frac{2}{p-\delta}\sum_{k=1}^{\infty}\Re\left(\left(\frac{1-\delta}{p-\delta}\right)^{\frac{2\pi i}{\ln(p)}} B\left(n, 1+\frac{2k\pi i}{\ln(p)}\right)\right) +$$

$$\frac{2}{p}\sum_{k=1}^{\infty}\Re\left(B\left(n, 1+\frac{2k\pi i}{\ln(p)}\right)\right).$$

# Unknown number of stations

## Upper approximation



Let $H(n) = T(n) + L(Y(n))$. Then

$$C(p, \delta, U) = \max_{m \le U} \frac{1}{E[Y(m)]} \cdot \max_{m \le U} E[H(m)]$$

**Theorem**

*Let U be an upper bound on a number of stations. Then the total number of slots is bounded by*

$$(1 + \mathcal{C}(p, \delta, U)) \cdot n.$$

# Unknown number of stations

## Upper approximation on C



### Theorem

$$C(p,\delta,U) \leq \frac{1}{\psi(p)}\left(W(\delta,p,U) + \frac{1}{2} + \frac{H_U}{\ln(1/p)}\right)$$

*where*

1. $\psi(p) = \frac{1-p}{p\ln(1/p)}\left(1 - 2\sqrt{\frac{2\pi^2}{\ln(1/p)\sinh(2\pi^2/\ln(1/p))}}\right)$,
2. $W(\delta,p,U) = \max_{m\leq U} E[L(Y(m))]$.

# Outline

# Unknown number of stations
## Comparison with simulations

Let $\mathcal{C}(p^*, \delta, U)) = \min_p \mathcal{C}(p, \delta, U)$.

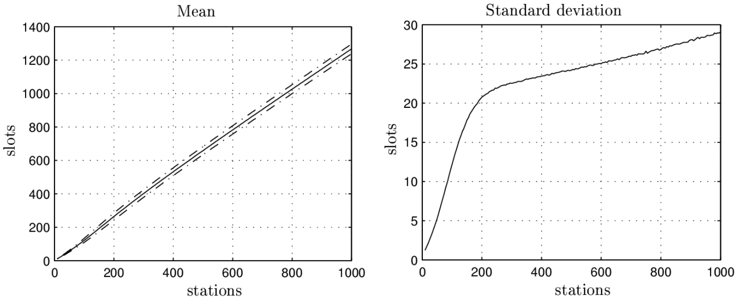
Table: Results for $\delta = 0.001$

| U | $p^*$ | $(1 + \mathcal{C}(p^*, \delta, U)) \cdot n$ | simulations |
|---|---|---|---|
| 100 | 0.037678 | $1.3271 \cdot n$ | $1.3168 \cdot n$ |
| 1000 | 0.0267521 | $1.3998 \cdot n$ | $1.3398 \cdot n$ |
| 10000 | 0.0232507 | $1.4677 \cdot n$ | $1.3482 \cdot n$ |

**Corollary**

Our estimation $\mathcal{C}(p, \delta, U)$ is very precise.

# Simulation results



## Comparison

CLW: Cai-Lu-Wang algorithm
CKZ: Cichon-Kutylowski-Zawada algorithm

Table: Results for $\lambda = 0.005$

| U | $p^*$ | CLW 2003 | CKZ 2006 |
|---|---|---|---|
| 100 | 0.0423848 | $\leq 1.6162 \cdot n$ | $\leq 1.5927 \cdot n$ |
| 1000 | 0.0267521 | $\leq 1.7497 \cdot n$ | $\leq 1.6381 \cdot n$ |
| 10000 | 0.0232507 | $\leq 1.9199 \cdot n$ | $\leq 1.7647 \cdot n$ |

# THANK YOU