Frameless ALOHA with latency-reliability guarantees

March 17, 2017

F. Lázaro ♣, Čedomir Stefanović †, Petar Popovski †

♣Institute of Communications and Navigation
German Aerospace Center (DLR)

† Department of Electronic Systems
Aalborg University

Random Access protocols

- random access (RA) protocols are effective when the nature of the traffic is unpredictable and/or when the number of active devices is dynamic and the total population very large

Random Access protocols

- random access (RA) protocols are effective when the nature of the traffic is unpredictable and/or when the number of active devices is dynamic and the total population very large
- the earliest examples of RA protocols are ALOHA [Abr-70] and Slotted ALOHA [Rob-75], which have been adopted in many standards (GSM, UMTS, DVB-RCS2, ...), typically for control channels only. Issues:

Random Access protocols

- random access (RA) protocols are effective when the nature of the traffic is unpredictable and/or when the number of active devices is dynamic and the total population very large
- the earliest examples of RA protocols are ALOHA [Abr-70] and Slotted ALOHA [Rob-75], which have been adopted in many standards (GSM, UMTS, DVB-RCS2, ...), typically for control channels only. Issues:

[Abr-70] N. Abramson "The ALOHA system: Another alternative for computer communications", in Proc. of the 1970 Fall Joint Comput. Conf. (AFIPS)

[Rob-75] L.G. Roberts "ALOHA Packet System with and without Slots and Capture", in Proc. SIGCOMM Comput. Commun. Rev., ACM, 1975

Random Access protocols

- random access (RA) protocols are effective when the nature of the traffic is unpredictable and/or when the number of active devices is dynamic and the total population very large
- the earliest examples of RA protocols are ALOHA [Abr-70] and Slotted ALOHA [Rob-75], which have been adopted in many standards (GSM, UMTS, DVB-RCS2, ...), typically for control channels only. Issues:
  - low peak throughput ($1/(2e)$ and $1/e$ respectively)

[Abr-70] N. Abramson “The ALOHA system: Another alternative for computer communications”, in Proc. of the 1970 Fall Joint Comput. Conf. (AFIPS)

[Rob-75] L.G. Roberts “ALOHA Packet System with and without Slots and Capture”, in Proc. SIGCOMM Comput. Commun. Rev., ACM, 1975

Random Access protocols

- random access (RA) protocols are effective when the nature of the traffic is unpredictable and/or when the number of active devices is dynamic and the total population very large
- the earliest examples of RA protocols are ALOHA [Abr-70] and Slotted ALOHA [Rob-75], which have been adopted in many standards (GSM, UMTS, DVB-RCS2, ...), typically for control channels only. Issues:
  - low peak throughput ($1/(2e)$ and $1/e$ respectively)
  - high packet loss rate even for low load

[Abr-70] N. Abramson "The ALOHA system: Another alternative for computer communications", in Proc. of the 1970 Fall Joint Comput. Conf. (AFIPS)

[Rob-75] L.G. Roberts "ALOHA Packet System with and without Slots and Capture", in Proc. SIGCOMM Comput. Commun. Rev., ACM, 1975

Advanced Random Access protocols

- recently it has been shown that much higher efficiency can be obtained by introducing time diversity (replication) and successive interference cancellation (SIC) at the receiver

Advanced Random Access protocols

- recently it has been shown that much higher efficiency can be obtained by introducing time diversity (replication) and successive interference cancellation (SIC) at the receiver
- examples:

Advanced Random Access protocols

- recently it has been shown that much higher efficiency can be obtained by introducing time diversity (replication) and successive interference cancellation (SIC) at the receiver
- examples:
  - CRDSA [Cas-07]
  - IRSA [Liva-11]

[Cas-07] E. Casini, R. De Gaudenzi, and O. del Rio Herrero, "Contention Resolution Diversity Slotted ALOHA (CRDSA): An Enhanced Random Access Scheme for Satellite Access Packet Networks", IEEE Trans. on Wireless Commun., vol. 6, no. 4, pp. 1408-1419, April 2007

[Liva-11] G. Liva, "Graph-Based Analysis and Optimization of Contention Resolution Diversity Slotted ALOHA", IEEE Trans. Commun., vol. 59, no. 2, pp. 477-487, Feb. 2011

Frameless ALOHA

- Frameless ALOHA is a slotted RA protocol that exploits ideas originating from the rateless coding framework:

Frameless ALOHA

- Frameless ALOHA is a slotted RA protocol that exploits ideas originating from the rateless coding framework:
  - the length of the contention period is not fixed a priori

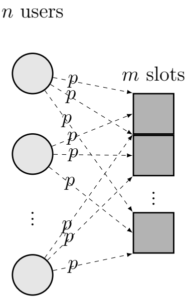
Frameless ALOHA

- Frameless ALOHA is a slotted RA protocol that exploits ideas originating from the rateless coding framework:
  - the length of the contention period is not fixed a priori
  - every user is active (sends a packet) in a slot with probability p

Frameless ALOHA

- Frameless ALOHA is a slotted RA protocol that exploits ideas originating from the rateless coding framework:
  - the length of the contention period is not fixed a priori
  - every user is active (sends a packet) in a slot with probability p

$$p = \frac{\beta}{n}$$



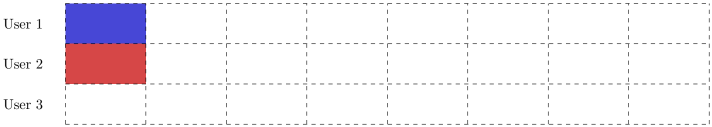
Frameless ALOHA
Example

contention starts

User 1								
User 2								
User 3								

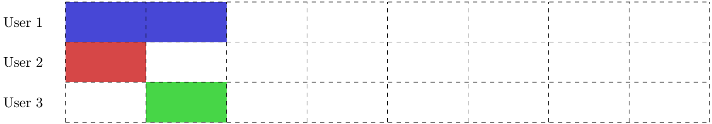
Frameless ALOHA
Example

slot 1



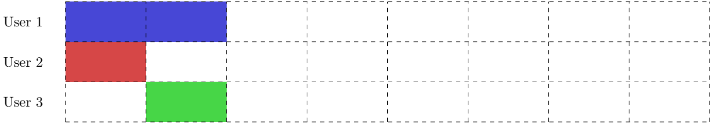
Frameless ALOHA
Example

slot 2



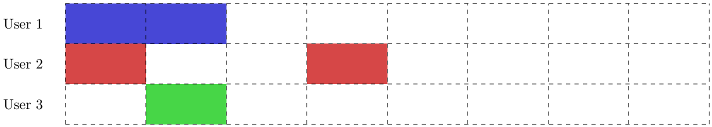
Frameless ALOHA
Example

slot 3



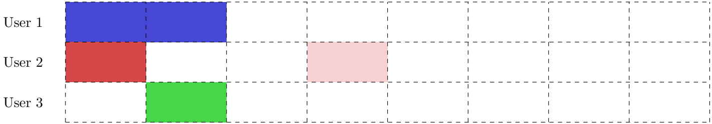
Frameless ALOHA
Example

slot 4



Frameless ALOHA
Example

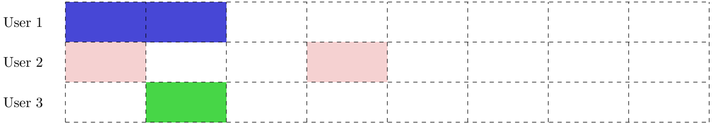
slot 4



decoding starts

Frameless ALOHA
Example

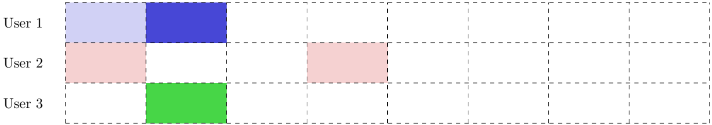
slot 4



decoding continues

Frameless ALOHA
Example

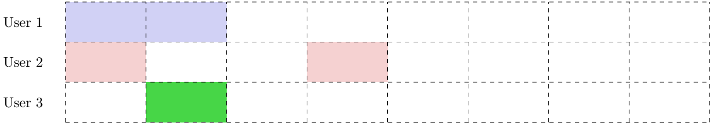
slot 4



decoding continues

Frameless ALOHA
Example

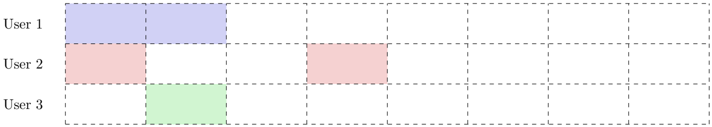
slot 4



decoding continues

Frameless ALOHA
Example

slot 4



since all users are recovered we can terminate the contention after 5 slots

Finite Length Analysis
Definitions

Definition (Ripple)

We define the ripple as the set of singleton slots (reduced degree 1) and we denote it by $\mathscr{R}$.

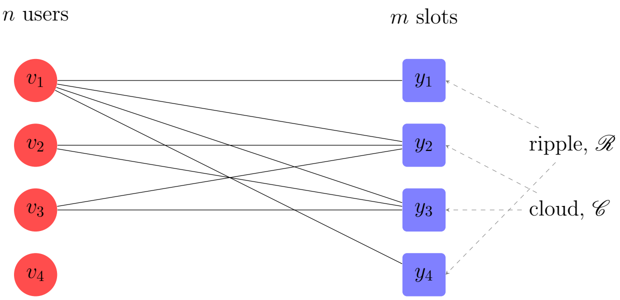
the cardinality of the ripple is denoted by r and its associated random variable as R.

Definition (Cloud)

We define the cloud as the set of slots with reduced degree $d \geq 2$ and we denote it by $\mathscr{C}$.

the cardinality of the cloud is denoted by c and the corresponding random variable as C.

Finite Length Analysis
Bipartite Graph Representation



Finite Length Analysis
Bipartite Graph Representation

- the slot degree distribution is given by

$$\Omega_i = \binom{n}{i} p^i (1-p)^{n-i}.$$

Finite Length Analysis
Bipartite Graph Representation

- the slot degree distribution is given by

$$\Omega_i = \binom{n}{i} p^i (1-p)^{n-i}.$$

- in the collision channel, the iterative SIC process can be seen as an iterative pruning of the bipartite graph.

Finite Length Analysis
Bipartite Graph Representation

- the slot degree distribution is given by

$$\Omega_i = \binom{n}{i} p^i (1-p)^{n-i}.$$

- in the collision channel, the iterative SIC process can be seen as an iterative pruning of the bipartite graph.
  - intially all n users are unresolved

Finite Length Analysis
Bipartite Graph Representation

- the slot degree distribution is given by

$$\Omega_i = \binom{n}{i} p^i (1-p)^{n-i}.$$

- in the collision channel, the iterative SIC process can be seen as an iterative pruning of the bipartite graph.
  - intially all n users are unresolved
  - at every iteration:

Finite Length Analysis
Bipartite Graph Representation

- the slot degree distribution is given by

$$\Omega_i = \binom{n}{i} p^i (1-p)^{n-i}.$$

- in the collision channel, the iterative SIC process can be seen as an iterative pruning of the bipartite graph.
  - intially all n users are unresolved
  - at every iteration:
    - if there are singleton slots $\rightarrow$ one user is resolved
    - if there are no singleton slots decoding stops

Finite Length Analysis
Finite State Machine

- the iterative SIC process can be modelled by means of a **finite state machine** with state:

$$\mathrm{S}_u := (\mathrm{C}_u, \mathrm{R}_u)$$

Finite Length Analysis
Finite State Machine

- the iterative SIC process can be modelled by means of a **finite state machine** with state:

$$\mathrm{S}_u := (\mathrm{C}_u, \mathrm{R}_u)$$

  - C_u: cardinality of cloud when u users are unresolved

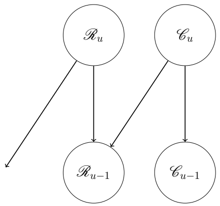
Finite Length Analysis
Finite State Machine

- the iterative SIC process can be modelled by means of a **finite state machine** with state:

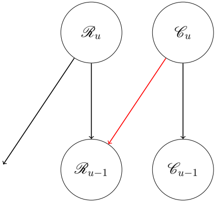
$$\mathtt{S}_u := (\mathtt{C}_u, \mathtt{R}_u)$$

  - $\mathtt{C}_u$: cardinality of cloud when u users are unresolved
  - $\mathtt{R}_u$: cardinality of ripple when u users are unresolved

Finite Length Analysis
Transition from u to $u-1$ resolved users



Finite Length Analysis
Transition from u to $u-1$ resolved users



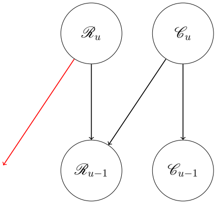
- $\mathtt{b}_u$: the number of slots leaving $\mathscr{C}_u$ and entering $\mathscr{R}_{u-1}$

$$\mathtt{b}_u := \mathtt{c}_u - \mathtt{c}_{u-1}$$

- B_u: associated random variable

Finite Length Analysis

Transition from u to $u-1$ resolved users



- $\mathtt{a}_u$: the number of slots leaving the ripple $\mathscr{R}_u$ in the transition
- $\mathtt{A}_u$: r.v. associated to $\mathtt{a}_u$

Finite Length Analysis
Main Result

Theorem

Given that the decoder is at state $\mathtt{S}_u = (\mathtt{c}_u, \mathtt{r}_u)$*, when* u *users are unresolved and with* $\mathtt{r}_u > 0$*, the probability of the decoder being at state* $\Pr\{\mathtt{S}_{u-1} = (\mathtt{c}_{u-1}, \mathtt{r}_{u-1})\}$ *when* $u-1$ *users are unresolved is given by*

$$\Pr\{\mathtt{S}_{u-1} = (\mathtt{c}_u - \mathtt{b}_u, \mathtt{r}_u - \mathtt{a}_u + \mathtt{b}_u) | \mathtt{S}_u = (\mathtt{c}_u, \mathtt{r}_u)\} = \binom{\mathtt{c}_u}{\mathtt{b}_u} {q_u}^{\mathtt{b}_u} (1-q_u)^{\mathtt{c}_u - \mathtt{b}_u} \binom{\mathtt{r}_u - 1}{\mathtt{a}_u - 1} \times \left(\frac{1}{u}\right)^{\mathtt{a}_u - 1} \left(1 - \frac{1}{u}\right)^{\mathtt{r}_u - \mathtt{a}_u}$$

for $0 \le \mathtt{b}_u \le \mathtt{c}_u$*,* $\mathtt{b}_u - \mathtt{a}_u \le \mathtt{r}_u$ *and* $\mathtt{a}_u \ge 1$*, and with*

$$q_u = \frac{\sum_{d=2}^{n-u-2} \Omega_d \, d(d-1) \frac{1}{n} \frac{u-1}{n-1} \frac{\binom{n-u}{d-2}}{\binom{n-2}{d-2}}}{1 - \sum_{d=1}^{n-u-1} \Omega_d \, u \frac{\binom{n-u}{d-1}}{\binom{n}{d}} - \sum_{d=0}^{n-u} \Omega_d \frac{\binom{n-u}{d}}{\binom{n}{d}}}.$$

Finite Length Analysis
Throughput and PER

- In practice one is interested in the packet error rate P_e

$$P_e = \sum_{u=1}^{n} \sum_{\mathsf{c}_u} \frac{u}{n} \Pr\{\mathsf{S}_u = (\mathsf{c}_u, 0)\}.$$

Finite Length Analysis
Throughput and PER

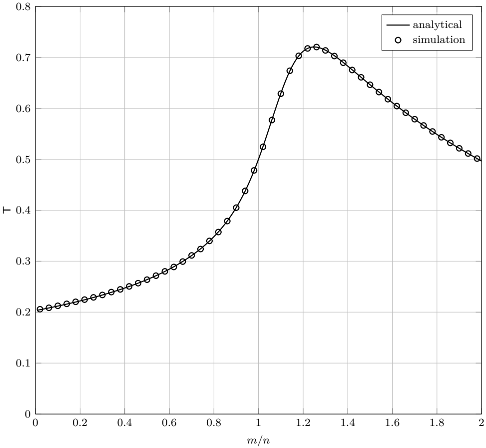
- In practice one is interested in the packet error rate $\mathsf{P_e}$

$$\mathsf{P_e} = \sum_{u=1}^{n} \sum_{\mathsf{c}_u} \frac{u}{n} \Pr\{\mathsf{S}_u = (\mathsf{c}_u, 0)\}.$$

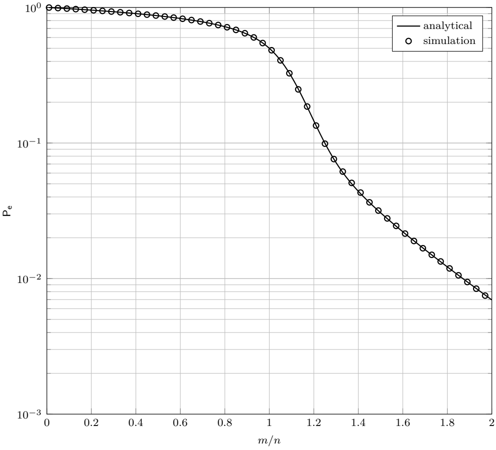
- The throughput T is the number of resolved users normalized by the number of slots:

$$\mathsf{T} = \frac{n(1-\mathsf{P_e})}{m} = \frac{1-\mathsf{P_e}}{m/n}.$$

Finite Length Analysis
Throughput for $\beta = 2.5$, for $n = 100$



Finite Length Analysis
PER for $\beta = 2.5$, for $n = 100$



Latency-Reliability Guarantees
Motivation

- the analysis provides not only the PER but it can provide the exact probability that k users are undecoded after m slots

Latency-Reliability Guarantees
Motivation

- the analysis provides not only the PER but it can provide the exact probability that k users are undecoded after m slots
  - it provides a latency-reliability guarantee

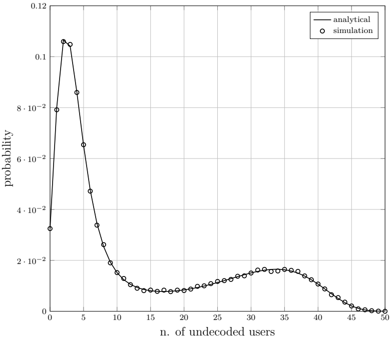
Latency-Reliability Guarantees

Motivation

- the analysis provides not only the PER but it can provide the exact probability that k users are undecoded after m slots
  - it provides a latency-reliability guarantee
  - in some applications one wants to decode at least k out of n with a very high probability (for example, vehicular networks or industrial automation)

Latency-Reliability Guarantees
Motivation

- the analysis provides not only the PER but it can provide the exact probability that k users are undecoded after m slots
  - it provides a latency-reliability guarantee
  - in some applications one wants to decode at least k out of n with a very high probability (for example, vehicular networks or industrial automation)
  - Example: $n = 50$ users, $m = 100$ slots



Latency-Reliability Guarantees
Optimization

- the analysis can be used to optimize the performance of frameless Aloha

Latency-Reliability Guarantees
Optimization

- the analysis can be used to optimize the performance of frameless Aloha
- we can perform the following optimization

Latency-Reliability Guarantees
Optimization

- the analysis can be used to optimize the performance of frameless Aloha
- we can perform the following optimization
  - maximize P_k the probability that at least k users out of n are decoded after a contention of m slots

Latency-Reliability Guarantees
Optimization

- the analysis can be used to optimize the performance of frameless Aloha
- we can perform the following optimization
  - maximize P_k the probability that at least k users out of n are decoded after a contention of m slots
- in this presentation we shall focus on the following scenario

Latency-Reliability Guarantees
Optimization

- the analysis can be used to optimize the performance of frameless Aloha
- we can perform the following optimization
  - maximize P_k the probability that at least k users out of n are decoded after a contention of m slots
- in this presentation we shall focus on the following scenario
  - number of users: $n = 50$

Latency-Reliability Guarantees
Optimization

- the analysis can be used to optimize the performance of frameless Aloha
- we can perform the following optimization
  - maximize P_k the probability that at least k users out of n are decoded after a contention of m slots
- in this presentation we shall focus on the following scenario
  - number of users: $n = 50$
  - reliability target: $k = n = 50$

Latency-Reliability Guarantees
Optimization

- the analysis can be used to optimize the performance of frameless Aloha
- we can perform the following optimization
  - maximize P_k the probability that at least k users out of n are decoded after a contention of m slots
- in this presentation we shall focus on the following scenario
  - number of users: $n = 50$
  - reliability target: $k = n = 50$
  - latency target (in slots): 100

Latency-Reliability Guarantees
Optimization

- the analysis can be used to optimize the performance of frameless Aloha
- we can perform the following optimization
  - maximize P_k the probability that at least k users out of n are decoded after a contention of m slots
- in this presentation we shall focus on the following scenario
  - number of users: $n = 50$
  - reliability target: $k = n = 50$
  - latency target (in slots): 100
- in this scenario the optimal parameter is $\beta = 3.33$, which leads to $P_k = 0.9334$

Latency-Reliability Guarantees
Optimization

- the analysis can be used to optimize the performance of frameless Aloha
- we can perform the following optimization
  - maximize P_k the probability that at least k users out of n are decoded after a contention of m slots
- in this presentation we shall focus on the following scenario
  - number of users: $n = 50$
  - reliability target: $k = n = 50$
  - latency target (in slots): 100
- in this scenario the optimal parameter is $\beta = 3.33$, which leads to $P_k = 0.9334$
- this value of P_k might be too high for many applications

Latency-Reliability Guarantees
Optimization II

- How can we improve the performance?

Latency-Reliability Guarantees
Optimization II

- How can we improve the performance?
- we can have different classes of slots

Latency-Reliability Guarantees
Optimization II

- How can we improve the performance?
- we can have different classes of slots
  - m_1 slots with β_1

Latency-Reliability Guarantees
Optimization II

- How can we improve the performance?
- we can have different classes of slots
  - m_1 slots with β_1
  - m_2 slots with β_2

Latency-Reliability Guarantees
Optimization II

- How can we improve the performance?
- we can have different classes of slots
  - m_1 slots with β_1
  - m_2 slots with β_2
  - ...
  - m_h slots with β_h

Latency-Reliability Guarantees
Optimization II

- How can we improve the performance?
- we can have different classes of slots
  - m_1 slots with β_1
  - m_2 slots with β_2
  - ...
  - m_h slots with β_h
- the analysis needs to be modified
  - the iterative SIC process is modelled by means of a finite state machine with state:
    $$\mathtt{S}_u := (\mathtt{C}_u^{(1)}, \mathtt{C}_u^{(2)}, \ldots, \mathtt{C}_u^{(h)}, \mathtt{R}_u)$$
  - $\mathtt{C}_u^{(i)}$: cardinality of i-th cloud when u users are unresolved (number of slots of type i with reduced degree 2 or larger)
  - $\mathtt{R}_u$: cardinality of ripple when u users are unresolved

Latency-Reliability Guarantees

Optimization - Results

- how much can we improve the performance?

Latency-Reliability Guarantees
Optimization - Results

- how much can we improve the performance?

slot classes	P_k	parameters
1	0.9334	$\beta = 3.33$, $m = 100$
2	0.9986	$\beta_1 = 2.53$, $m_1 = 86$ $\beta_2 = 22.08$, $m_2 = 14$
3	0.99917	$\beta_1 = 2.51$, $m_1 = 88$ $\beta_2 = 17.39$, $m_2 = 11$ $\beta_3 = 50$, $m_3 = 1$
dynamical	0.9999975	dynamical

Latency-Reliability Guarantees
Optimization - Results

- how much can we improve the performance?

slot classes	P_k	parameters
1	0.9334	$\beta = 3.33$, $m = 100$
2	0.9986	$\beta_1 = 2.53$, $m_1 = 86$ $\beta_2 = 22.08$, $m_2 = 14$
3	0.99917	$\beta_1 = 2.51$, $m_1 = 88$ $\beta_2 = 17.39$, $m_2 = 11$ $\beta_3 = 50$, $m_3 = 1$
dynamical	0.9999975	dynamical

- dynamical means that using feedback β is varied on a slot basis depending on the decoder state:

Latency-Reliability Guarantees

Optimization - Results

- how much can we improve the performance?

slot classes	P_k	parameters
1	0.9334	$\beta = 3.33$, $m = 100$
2	0.9986	$\beta_1 = 2.53$, $m_1 = 86$ $\beta_2 = 22.08$, $m_2 = 14$
3	0.99917	$\beta_1 = 2.51$, $m_1 = 88$ $\beta_2 = 17.39$, $m_2 = 11$ $\beta_3 = 50$, $m_3 = 1$
dynamical	0.9999975	dynamical

- dynamical means that using feedback β is varied on a slot basis depending on the decoder state:
  - the number of slots in each cloud type $c_u^{(i)}$

Latency-Reliability Guarantees
Optimization - Results

- how much can we improve the performance?

slot classes	P_k	parameters
1	0.9334	$\beta = 3.33,\ m = 100$
2	0.9986	$\beta_1 = 2.53,\ m_1 = 86$ $\beta_2 = 22.08,\ m_2 = 14$
3	0.99917	$\beta_1 = 2.51,\ m_1 = 88$ $\beta_2 = 17.39,\ m_2 = 11$ $\beta_3 = 50,\ m_3 = 1$
dynamical	0.9999975	dynamical

- dynamical means that using feedback β is varied on a slot basis depending on the decoder state:
  - the number of slots in each cloud type $\mathrm{c}_u^{(i)}$
  - the number of decoded users

Conclusions

- our work is based on an **exact finite-length analysis** of frameless ALOHA in the collision channel

Conclusions

- our work is based on an **exact finite-length analysis** of frameless ALOHA in the collision channel
- the analysis yields the exact probability of meeting a **latency-reliability** target

Conclusions

- our work is based on an **exact finite-length analysis** of frameless ALOHA in the collision channel
- the analysis yields the exact probability of meeting a **latency-reliability** target
- application example: introducing different slot classes we can decrease the probability of not meeting a latency-reliability target by almost 2 orders of magnitude

Conclusions

- our work is based on an **exact finite-length analysis** of frameless ALOHA in the collision channel
- the analysis yields the exact probability of meeting a **latency-reliability** target
- application example: introducing different slot classes we can decrease the probability of not meeting a latency-reliability target by almost 2 orders of magnitude
- a dynamical strategy can provide even further gains (2 orders of magnitude better than static).