Random Multiple Access in q-ary Disjunctive Channel

Eugeny Pustovalov and Andrey Turlikov,
State University of Aerospace Instrumentation,
St. Petersburg, RUSSIA
E-mails: eugeny@vu.spb.ru, turlikov@vu.spb.ru

***Abstract*—Random Multiple Access (RMA) in q-ary disjunctive channel is considered. Novel RMA algorithm is suggested. The algorithm belongs to the class of splitting algorithms. It reduces number of retransmissions in q-ary disjunctive channel due to exploiting properties of the channel. The algorithm has throughput of 0.603 which is grater than throughput of conventional RMA algorithms that do not exploit properties of considered channel.**

***Index Terms*—random multiple access, channel without intensity information, maximal stable throughput**

I. Introduction

Random Multiple Access (RMA) is preferred channel access method for the low intensity traffic where multiple packets rarely queue up at one transmitter. For such type of traffic RMA algorithms ensure low mean packet delay even for the considerably high user population. Mean packet delay provided by RMA algorithms rises with increase of overall traffic intensity and the algorithms could not ensure finite delay when arrival rate reaches the limit called maximal stable throughput (MST).

Conventional RMA algorithms, such as ALOHA, Binary exponential backoff, Tree-algorithms [1], are realized under a standard information-theoretical model [2], [3], assuming that if several users simultaneously transmit their packets, the information will be corrupted due to interference and no packets could be received correctly. Such event is called collision. To resolve the collisions, all collided packets must be retransmitted according to the algorithms. Throughput bound of standard RMA model is 0.567 [4], while the highest known MST is 0.4877 of First Come First Serve (FCFS) algorithm [5].

Recently a novel RMA model with Successive Interference Cancellation (SIC) was introduced. When collision is indicated, received signal is stored in the receivers memory. When users retransmit their data, the stored signal is processed in the way to restore the rest packets. This procedure is called Successive Interference Cancellation. SIC could potentially reduce number of retransmission and this way increase MST. Successive interference cancellation tree algorithm (SICTA) which uses the SIC mechanism was designed in [6] and shown to achieve an MST as high as 0.693. While SICTA achieves MST that is higher than throughput bound of conventional RMA algorithms, the algorithm is primary theoretical one because it does not define how to realise SIC and assumes that SIC could always restore user data without errors. In [7] the modified SICTA (R-SICTA) algorithm was suggested that is robust to potential SIC errors and restrictions on receiver memory. But SIC realisation and channel model were not considered either. Finally, in [8] an FCFS-type algorithm with SIC was introduced and analysed in the same theoretical model.

In this paper we consider random multiple access in q-ary disjunctive channel. The channel was studied in [9] where it was called T-user M-frequency channel without intensity information. Although the channel was initially defined in terms of frequencies, it was mentioned that it corresponds to any signaling scheme with M orthogonal signals, e.g. pulse position modulation (PPM). Here we do not study the physical nature of the channel and following [9] consider idealistic noiseless channel model where one of q orthogonal signals could be transmitted in each signaling interval (time slot). For such channel a novel RMA algorithm is suggested with MST (0.603) grater than MST of conventional algorithms.

The rest of the paper is organized as follows. In Section II channel model and list of assumptions are defined. Section III discusses how to extract data packets from the channel output. And in Section IV instruction of novel RMA algorithm are introduced.

II. System model and assumptions

Random packet access over a common channel is considered. System model is defined by the following assumptions.

1) Discrete time system is considered. Transmission time is divided into frames. Each frame is divided into N slots. In each slot a user could transmit one of q orthogonal signals (one q-ary symbol). User packet consists of N q-ary symbols and is transmitted within the frame. First symbol of the packet is transmitted in the first slot of the frame.
2) User device is shown in Figure 1. Transmission and reception operations are controlled by two algorithms: RMA algorithm and channel vector processing algorithm. The RMA algorithm (see section IV) indicates frames where user should transmit packet to the channel, including retransmissions. The channel vector processing algorithm (see section III) monitors channel state and extracts data packets from the channel output vector $\boldsymbol{Y}_t$ received at the end of each frame. The channel

processing algorithm feedbacks channel state θ_t to the RMA algorithm.

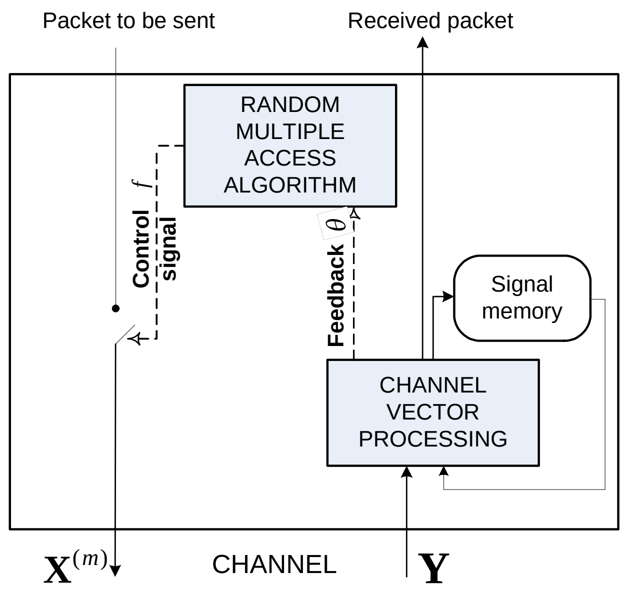

Figure 1. User device

3) Denote by $\boldsymbol{X}_t^{(m)}$ the packet of m-th user in t-th frame.

$$\boldsymbol{X}_t^{(m)} = \left(x_{tN}^{(m)}, x_{tN+1}^{(m)}, ..., x_{tN+N-1}^{(t)}\right),$$

where

$$x_n^{(m)} \in \{0, 1, ..., q-1\}, \ n = 0..N-1.$$

Let K users simultaneously transmit their packets in frame t . Channel output in each n-th slot of the frame t is K-size vector

$$\boldsymbol{x}_n = \begin{pmatrix} x_n^{(1)} \\ x_n^{(2)} \\ ... \\ x_n^{(K)} \end{pmatrix}, \qquad (1)$$

where each element of $\boldsymbol{x}_n$ is one q-ary symbols. Channel output in the n-th slot of the frame t is q-size vector

$$\boldsymbol{y}_n = (y_{n,0}, y_{n,1}, ..., y_{n,q-1}), \qquad (2)$$

where each element of $\boldsymbol{y}_n$ takes one of two binary values

$$y_{n,j} = \begin{cases} 0, & \text{if } x_n^{(m)} \neq j, \ \forall m = 1..K, \\ 1, & \text{else.} \end{cases} \qquad (3)$$

The model defined by (1)-(3) is called q-ary disjunctive channel (In [9] the same model is called q-frequency channel without intensity information or A-channel).

4) At the end of current frame t the user has reliable information about channel output $\boldsymbol{Y}_t$ in all N slots of this frame

$$\boldsymbol{Y}_t = \left(\boldsymbol{y}_{tN}, \boldsymbol{y}_{tN+1}, ..., \boldsymbol{y}_{tN+N-1}\right).$$

The $\boldsymbol{Y}_t$ is said to be channel vector.

5) Each packet has a label which unambiguously identifies the sender. Packets of different users differ at least in one symbol. The packet is not changed due to retransmissions. Each packet has a checksum that could reliably identify the valid packet for any changes in the data.

III. Channel vector processing

A. Channel state monitoring

Using the model introduced in Section II consider possible channel states in a frame. In the frame t the user receives N-size vector $\boldsymbol{Y}_t$, where each element of $\boldsymbol{Y}_t$ is q-size vector $\boldsymbol{y}_n$. Elements of $\boldsymbol{y}_n$ are calculated via (3). By $w(\boldsymbol{y})$ denote weight of the binary vector $\boldsymbol{y}$. Consider possible events in the frame t.

If no users transmit in the frame than

$$w(\boldsymbol{y}_n) = 0, \ \forall \ n = 0..N-1.$$

This event is called "idle".

If only one user transmits in the frame than

$$w(\boldsymbol{y}_n) = 1, \ \forall \ n = 0..N-1,$$

and unity in each $\boldsymbol{y}_n$ is placed in the position corresponding to the transmitted symbol according to (3). Such event is called "success".

Finally, if K users transmit in the frame ($K \geq 2$) than

$$1 \leq w(\boldsymbol{y}_n) \leq \max(K, q), \ \forall \ n = 0..N-1.$$

Consider possible values of $\boldsymbol{y}_n$. If all users transmit the same q-ary symbol s in the slot n than

$$\begin{cases} w(\boldsymbol{y}_n) = 1 \\ y_{n,s} = 1 \end{cases},$$

in other words, vector $\boldsymbol{y}_n$ has only single "1" at the position corresponding to transmitted symbol q-ary symbol. In this case the receiver could reliably identify the transmitted symbol.

If at least two users transmit different symbols in n-th slot than

$$w(\boldsymbol{y}_n) > 1.$$

In this case the receiver gets $w(\boldsymbol{y}_n)$ q-ary symbols and could not distinguish what packet each symbol belongs to. This leads to ambiguity in the slot. According to assumption 4 two and more simultaneously transmitted packets differ at least in one symbol. Thus, if two ore more users transmit their packets in the same frame, the receiver could not correctly receive any packet. This event is called "collision".

Conventional RMA model requires retransmission of each collided packet. In the considered channel model it is possible to retransmit some collided packets and extract the rest packet via special procedure called "collision compensation". This procedure is described below.

B. Collision compensation for the case of 2 packets

Consider collision of two packets ($K = 2$). For simplicity, let us start from the case of $q = 2$, which means that each packet consist of binary symbols (0 or 1). Vector $\boldsymbol{Y}_t$ received from the channel in the frame t has the following elements $\boldsymbol{y}_n$:

$$\boldsymbol{y}_n = \begin{cases} (1,0), & \text{if } x_n^{(1)} = x_n^{(2)} = 0, \\ (0,1), & \text{if } x_n^{(1)} = x_n^{(2)} = 1, \\ (1,1), & \text{if } x_n^{(1)} \neq x_n^{(2)}. \end{cases}$$

Retransmissions are required to resolve the ambiguous elements in $\boldsymbol{Y}_t$. Let 1st user successfully retransmits its packet $\boldsymbol{X}^{(1)}$ in the frame $t+m$. According to assumption 4 (see section II) the user packet is not changed within retransmissions. Having vector $\boldsymbol{Y}_t$ and packet $\boldsymbol{X}^{(1)}$ the receiver could reconstruct symbols of the second packet $\boldsymbol{X}^{(2)}$ as follows

$$x_n^{(2)} = \begin{cases} x_n^{(1)}, & \text{if } w(\boldsymbol{y}_n) = 1, \\ \neg x_n^{(1)}, & \text{if } w(\boldsymbol{y}_n) = 2, \end{cases} \tag{4}$$

where $\neq$ denotes binary inversion.

Concluding, to resolve collision of two packets only one of collided packets should be retransmitted. The same statement is valid for the case of arbitrary q-ary signals ($q > 2$). The collision resolution algorithm for $q > 2$ operates as follows:

1) For each $\boldsymbol{y}_n$ construct set $\boldsymbol{b}_n$ with elements i such as

$$i \in \boldsymbol{b}_n \Leftrightarrow y_{n,i} = 1.$$

2) Calculate symbols of the second packet as

$$x_n^{(2)} = \begin{cases} x_n^{(1)}, & \text{if } w(\boldsymbol{y}_n) = 1, \\ \boldsymbol{b}_n \backslash x_n^{(1)}, & \text{if } w(\boldsymbol{y}_n) = 2. \end{cases} \tag{5}$$

C. Collision resolution for the case of collision multiplicity $K > 2$

Consider collision of tree packets ($K = 3$) for the case of binary transmission ($q = 2$). Vector $\boldsymbol{Y}_t$ received from the channel in the frame t has the following elements $\boldsymbol{y}_n$:

$$\boldsymbol{y}_n = \begin{cases} (1,0), & \text{if } x_n^{(1)} = x_n^{(2)} = x_n^{(3)} = 0, \\ (0,1), & \text{if } x_n^{(1)} = x_n^{(2)} = x_n^{(3)} = 1, \\ (1,1), & \text{else}. \end{cases}$$

Consider collision resolution procedure. Let 1st and 2nd packets have been retransmitted correctly. The question is if the rest packet could be reconstructed similar to the case of two-packets collision? Consider n-th slot in the frame. If n-th bits of the 1st and 2nd packets are identical, the restoration of the n-th bit of the 3rd packet is identical to the case of two packets. If n-th bits of the 1st and 2nd packets are different (e.g. $x_n^{(1)} = 1$ and $x_n^{(1)} = 0$), than single channel output vector $\boldsymbol{y}_n = (1,1)$ corresponds to $x_n^{(3)} = 0$ as well as $x_n^{(3)} = 1$. This means, that collision of 3 packets requires retransmission of all collided packets. By induction the same statement is valid for any arbitrary number of packets in collision.

D. General algorithm of channel vector processing

In Sections III-B-III-C has been shown that collision of two packets requires only one retransmission while collision of more than two packets requires retransmission of all collided packets. Note than channel output $\boldsymbol{Y}$ does not allow evolution of collision multiplicity due to disjunctive channel property. Also remind that according to assumption 4 the user packet contains checksum that could reliably identify the valid packet. Thus if collision compensation procedure is failed due to high number of packets in collision, it can be revealed via checksum. Finally, the following channel vector processing algorithm could be suggested

1) Determine channel state event (empty, success or collision) based on received channel output vector $\boldsymbol{Y}$.
2) If collision revealed, save vector $\boldsymbol{Y}$ in the memory. If memory is not empty, the previous signal is deleted from the memory. Thus, in each frame the memory stores at most one vector $\boldsymbol{Y}$. Feedback control signal $\boldsymbol{c}$ to the RMA algorithm.
3) If valid user packet has been received (”success” event) and memory is not empty, apply collision compensation procedure to the stored vector via (4) or (5). Verify checksum of restored vector. If the checksum is correct than valid packet has been restored. If the checksum is not correct than vector $\boldsymbol{Y}$ corresponds to a collision of more than two packets and no data could be restored from it. Feedback signal $\boldsymbol{ss}$ to the RMA algorithm in the case of correct checksum or signal $\boldsymbol{sc}$ in the case of incorrect checksum. Empty memory in both cases.
4) If received vector corresponds to a valid user packet (”success” event) and memory is empty, feedback control signal $\boldsymbol{s}$ to the RMA algorithm.
5) If the frame is idle, feedback control signal $\boldsymbol{e}$ to the RMA algorithm.

IV. RMA ALGORITHM

In section III the algorithm which reconstructs user packets from the channel output vector was considered. Here we consider RMA algorithm, which controlls transmission process. Taking θ_t feedback from the channel vector processing algorithm, the RMA algorithm indicates frames where user packet should be transmitted. Following [5] let us give the formal definition of RMA algorithm.

Let new packets arrive in system according to Poisson process with intensity λ. Each station remembers time instant z of arriving its last packet and stores it until the packet is successfully transmitted.

In the frame t feedback θ_t might be one of the following values

$$\theta_t = \begin{cases} \boldsymbol{c}, & \text{collision}, \\ \boldsymbol{e}, & \text{idle frame}, \\ \boldsymbol{s}, & \text{success and no restoration attempts} \\ \boldsymbol{ss}, & \text{success and the second packet restored}, \\ \boldsymbol{sc}, & \text{success and collision compensation failed}. \end{cases} \tag{6}$$

The rules of θ_t calculation are defined in Section III-D.

Sequence $\boldsymbol{\theta}(t) = (\theta_0, ..., \theta_t)$ is called channel history by the frame t. It is supposed that at the beginning of frame $t+1$ all users know channel history $\boldsymbol{\theta}(t)$.

For the packet arrived at time instant z the user remembers sequence $\boldsymbol{v}^{(z)}(t) = \left(v_0^{(z)}, ..., v_t^{(z)}\right)$, where

$$v_i^{(z)} = \begin{cases} 0, & \text{if this packet was not transmitted in frame } i, \\ 1, & \text{if this packet was transmitted in frame } i. \end{cases}$$

The RMA algorithm is defined as a function $f\left(z, \boldsymbol{\theta}(t), \boldsymbol{v}^{(z)}(t)\right)$ with values in the interval $[0,1]$; its value is probability that a packet generated at time instant z will be transmitted in the frame t.

Let packet arrived at time instant z_n is successfully transmitted at time instant w_n. Value $\delta_n = w_n - z_n$ is called delay of n-th packet. δ_n is random value, which distribution depends on f. The algorithm f is said to be stable if

$$\lim_{n\to\infty} \Pr\{\delta_n < \infty\} = 1.$$

Maximal arrival intensity, implying that the given algorithm is stable, is called maximal stable throughput (MST).

Conventional RMA algorithms [2], [3], [5] use ternary feedback $\theta_t \in \{\boldsymbol{c}, \boldsymbol{s}, \boldsymbol{e}\}$. If feedback indicates collision, all collided packets are retransmitted according to RMA rules. Collision compensation procedure described in Section III could potentially avoid some retransmissions due to exploiting properties of the channel. However collision compensation does not give any benefits if conventional RMA algorithms with ternary feedback are used for collision resolution. To exploit benefits of collision compensation procedure, feedback θ_t should be extended to 5 valued according to (6) and RMA rules should be modified to take extended feedback into account.

Among conventional ternary-feedback RMA algorithms the most efficient one is First Come First Serve (FCFS) algorithm with an MST of 0.4877 [5]. Here we have modified FCFS protocol to take extended quinary feedback into account and reduce number of retransmissions.

The FCFS algorithm belongs to a class of windowed access RMA algorithms. In any frame t each station calculates two variables: $T(t)$ and $w(t)$, indicating a start of a current window and current window size correspondingly. All packets arrived withing interval $[T(t), T(t) + w(t))$ are transmitted in frame t. An initial window size is denoted as α and is a parameter of the algorithm.

Collision resolution procedure can be represented by a truncated binary tree called splitting tree. The examples of splitting trees for suggested FCFS algorithm are shown in Figure 2. Solid notes corresponds to actual frames where users transmit their packets. Dotted (skipped) nodes corresponds to virtual skipped frames where no transmission is required due to collision compensation. The node is skipped from the tree if $\boldsymbol{ss}$ or $\boldsymbol{sc}$ feedback signal is received in the previous frame. The original FCFS algorithm has the same tree, except that dotted nodes are not skipped and frames for transmission of corresponding packets are required. Thus the collision resolution duration of the novel algorithm is less than original one.

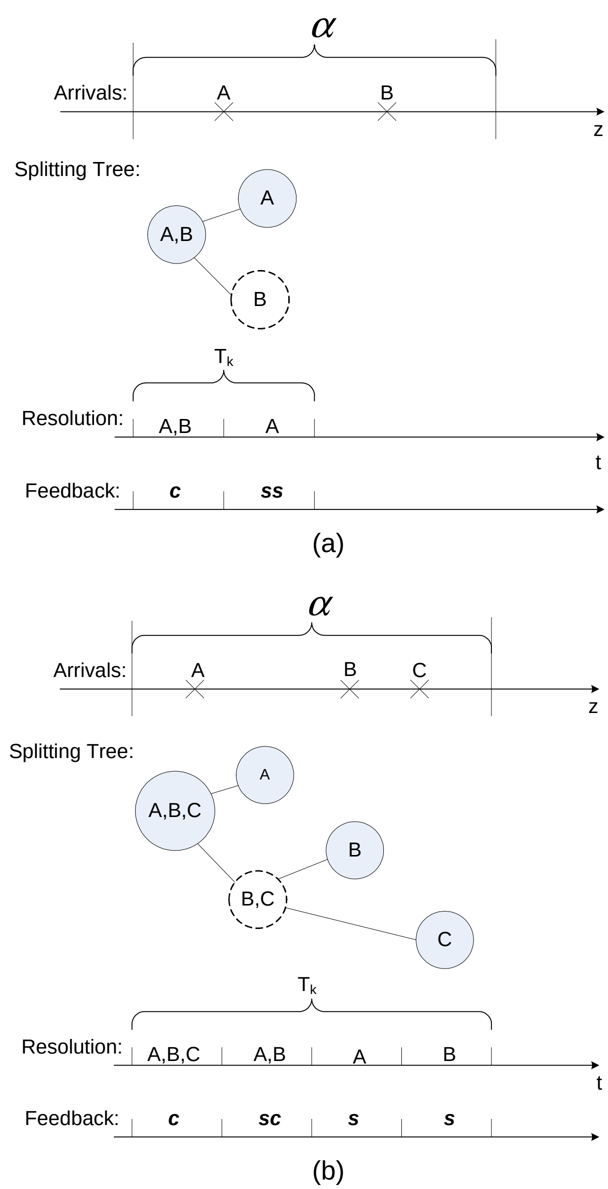

Figure 2. Illustration of suggested FCFS algorithm

Let us give the formal definition of suggested algorithm f. The algorithm takes at the input the time instant z of packet arrival, channel history $\boldsymbol{\theta}(t)$ and parameter α. The algorithm does not depend on $\boldsymbol{v}^{(z)}(t)$ and its output is binary: 0 or 1. In addition to variables T and w, the function f also uses auxiliary variable l which takes binary values (L or R). Algorithm pseudocode is shown in Figure 3.

Analysis of the algorithm is identical to [5] except that nodes skipping should be taken into account. Using technique from [5] we got MST $= 0.603$ of suggested algorithm.

```
function f(z, θ(t), α)
    T ← 0, w ← α, l ← R

    for i = 0,t do
        if θ_i = c then
            w ← w/2
            l ← L
        else if θ_i = e then
            T ← T + w
            if l = L then
                w ← w/2
        else if θ_i = ss then
            l ← R
            T ← T + 2w
            w ← α
        else if θ_i = sc then
            l ← L
            w ← w/2
        else if θ_i = s then
            T ← T + w
            if l = L then
                l ← R
            else
                w ← α

    if T < z < T + w then
        return 1                          ▷ Transmit packet
    else
        return 0                          ▷ Do not transmit packet
```

Figure 3. Suggested FCFS algorithm

V. Conclusion

In this paper the novel RMA algorithm in q-ary disjunctive channel has been introduced. Exploiting properties of the channel the algorithm reduces number of retransmission in collision resolution procedure in comparison to conventional RMA algorithms and this way reaches MST 0.603 which is grater than MST bound of conventional algorithms (0.567). The collision resolution procedure of suggested algorithm consists of two parts: channel vector processing algorithm which extracts data packets from the channel output and RMA instructions indicating frames where user must retransmit its packet. In real systems the channel vector processing algorithm could be implemented in physical layer of OSI model while RMA procedure works on Medium Access Control (MAC) sublayer.

It should be noted that MST of suggested algorithm does not depend on the signal set size q. Meanwhile it is know that throughput of q-ary disjunctive channel grows with increase of q and asymptotically reaches $0.693q$ [10]. High data rates in such channel could be achieved via coding. More over, in [11] RMA algorithms were designed assuming that any $N-1$ simultaneously transmitted packets do not collide due to coding (for arbitrary $N > 2$). MST of such algorithms grows with increase of minimal collision multiplicity N. Thus the problem of joint consideration of RMA and coding in disjunctive channel seems to be important and authors consider it as a way of further research.

References

[1] B. Tsybakov, "Survey of USSR contributions to random multiple-access communications," *IEEE Trans. Inform. Theory*, vol. 31, no. 2, pp. 143 – 165, 1985.

[2] B. S. Tsybakov and V. A. Mikhailov, "Free synchronous packet access in a broadcast channel with feedback," *Problems of Information Transmission*, vol. 14, no. 4, pp. 32–59, 1978.

[3] J. I. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Trans. Inform. Theory*, vol. 25, no. 5, pp. 505–515, 1979.

[4] B. S. Tsybakov and N. B. Likhanov, "Upper bound on the capacity of a random multiple-access system," *Problems of Information Transmission*, vol. 23, no. 3, pp. 64–78, 1987.

[5] B. S. Tsybakov and V. A. Mikhailov, "Random multiple packet access: Part-and-try algorithm," *Problems of Information Transmission*, vol. 16, no. 4, p. 6579, 1980.

[6] Y. Yingqun and G. B. Giannakis, "High-throughput random access using successive interference cancellation in a tree algorithm," *IEEE Trans. Inform. Theory*, vol. 53, no. 12, pp. 4628–4639, 2007.

[7] S. Andreev, E. Pustovalov, and A. Turlikov, "SICTA modifications with single memory location and resistant to cancellation errors," *Proc. of the 8th International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking NEW2AN 08*, p. 1324, 2008.

[8] B. V. Houdt and G. Peeters, "FCFS tree algorithms with interference cancellation and single signal memory requirements," *Proc. of International Conference on Telecommunications, ICT'08*, pp. 1 – 6, 2008.

[9] S. C. Chang and J. K. Wolf, "On the T-user M-frequency noiseless multiple access channel with and without intensity information," *IEEE Trans. Inform. Theory*, vol. 27, no. 1, pp. 41–48, 1981.

[10] L. Wilhelmsson and K. S. Zigangirov, "On the asymptotic capacity of a multiple-access channel," *Problems of Information Transmission*, vol. 33, no. 1, pp. 12–20, 1997.

[11] B. S. Tsybakov, V. A. Mikhailov, and N. B. Likhanov, "Bounds for packet transmission rate in a random-multiple-access system," *Problems of Information Transmission*, vol. 19, no. 1, p. 5068, 1983.