# Compute-and-Forward on a Line Network with Random Access

Zhijie Ren[1], Jasper Goseling[2,1], Jos Weber[1] and Michael Gastpar[3,1],
[1]Department of Intelligent Systems,
Delft University of Technology, The Netherlands
[2]Stochastic Operations Research,
University of Twente, The Netherlands
[3]Laboratory for Information in Networked Systems,
Ecole Polytechnique Fédérale de Lausanne, Switzerland
z.ren@tudelft.nl, j.goseling@utwente.nl, j.h.weber@tudelft.nl, michael.gastpar@epfl.ch

**Abstract**

Signal superposition and broadcast are important features of the wireless medium. Compute-and-Forward, also known as Physical Layer Network Coding (PLNC), is a technique exploiting these features in order to improve performance of wireless networks. More precisely, it allows wireless terminals to reliably decode a linear combination of all messages, when a superposition of the messages is received through the physical medium.

In this paper, we propose a random PLNC scheme for a local interference line network in which nodes perform random access scheduling. We prove that our PLNC scheme is capacity achieving in the case of one symmetric bi-directional session with terminals on both ends of this line network model. We demonstrate that our scheme significantly outperforms any other scheme. In particular, by eligibly choosing the access rate of the random access scheduling mechanism for the network, the throughput of our PLNC scheme is at least 3.4 and 1.7 times better than traditional routing and plain network coding, respectively.

# 1 Introduction

Compute-and-Forward (CF), also known as Reliable Physical Layer Network Coding (PLNC) [1], is a novel Network Coding (NC) technique for wireless networks. Uncoded versions of PLNC have been considered in the literature, see e.g. [2, 3], suffering from noise accumulation along the stages of the network. By contrast, CF works with codes in such a way as to allow nodes to efficiently and reliably recover a function of the messages from multiple senders. The technique of CF shows huge improvements over traditional routing and plain NC on many network models. Among them, the line network model has been extensively studied. Most of this work has focused on deterministic, centralized scheduling [2, 4, 5]. On the other hand, for plain NC, various network models with decentralized scheduling (e.g. random access) have been studied in [6, 7, 8], where various random NC schemes have been proposed.

In this paper, we consider CF for a line network with decentralized scheduling. In particular, we consider a random access scheduling mechanism. We propose a decentralized random PLNC scheme, which is proved to be feasible and optimal in the random access scenario. The throughput of our random PLNC scheme is compared to that of traditional routing and plain NC, where a significant enhancement of performance shows up. In particular, the throughput of our random PLNC scheme is at least 3.4 and 1.7 times better than traditional routing and plain NC, respectively. These improvements are obtained by optimizing the access rate of the random access scheduling mechanism individually for each scheme. If we use the same access rate for all schemes and let the rate approach 1, the improvement factor of the throughput of CF over the traditional routing and plain NC is even approaching infinity. This improvement is remarkably higher than the factors of 2 and 1.5, achieved by applying CF in the case of centralized scheduling.

The coding scheme that we introduce borrows some elements from the coding schemes for the line network as suggested by Pakzad et al. [9]. We will demonstrate that our coding scheme induces a network model that is within the scope of the work of Lun et al. [10]. Therefore, we will be able to resort to the maximum achievable throughput results of [10].

The paper is organized as follows. The model of a line network with random access is specified in Section 2. In Section 3, we propose our random PLNC scheme which can be embedded into the line network model with random access. Furthermore, in Section 4, the optimality of this scheme is proved by giving the capacity of the model and showing the capacity is achievable by our scheme. The comparison of our scheme to traditional routing and plain NC will also be given in Section 4 with some plots. Last but not least, we conclude our work and give our recommendations in Section 5.

# 2 Model

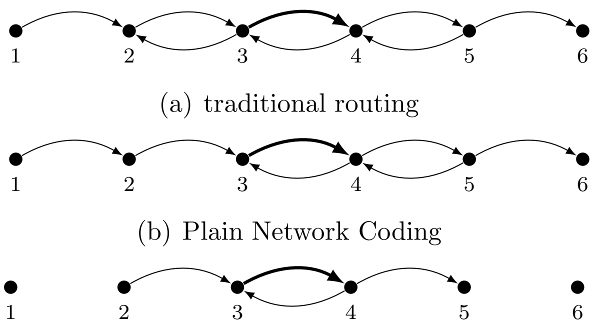
We consider a line topology modeled as a directed graph $(\mathcal{V}, \mathcal{E})$, with nodes $\mathcal{V} = \{1, 2, \ldots, M\}$ with unit distance, and edges $\mathcal{E} = \{(u, v) | u, v \in \mathcal{V}, |u - v| = 1\}$. Then, we build a communication model upon this topology, by considering the nodes in $\mathcal{V}$ as wireless devices. We assume time is slotted and transmitted messages are symbols from $\mathbb{F}(q) \cup \sigma$, where $\sigma$ denotes an empty transmission. Let $X_t(u)$ and $Y_t(u)$ denote the transmitted and received messages, respectively, for node $u$ in time slot $t$, and $A_t(u, v)$ the transmitted message on the directed edge $(u, v)$ in time slot $t$. The capacity of each edge is one symbol per time slot. We assume half-duplex constraints, i.e., a node cannot both transmit and receive in the same time slot. If node $u$ is not transmitting in time slot $t$ then $X_t(u) = \sigma$. For notation convenience, we sometimes use the symbol $\tau$ to denote a uniformly distributed random variable from $\mathbb{F}_q$ which is useless to the receiving node. We consider a local interference model in which the broadcast and superposition properties are now characterized as follows. For convenience, we define the nodes $\{m | m \leq 0 \vee m \geq M + 1\}$ as virtual nodes which are always silent.

**No Broadcast:** For any $u$ and $t$ it holds that

$$\begin{aligned} &A_t(u, u-1) = A_t(u, u+1) = X_t(u) = \sigma \text{ or} \\ &A_t(u, u-1) = X_t(u) \neq \sigma \text{ and } A_t(u, u+1) = \tau \text{ or} \\ &A_t(u, u+1) = X_t(u) \neq \sigma \text{ and } A_t(u, u-1) = \tau. \end{aligned} \tag{1}$$

**Broadcast:** For any $u$ and $t$ it holds that

$$A_t(u, u-1) = A_t(u, u+1) = X_t(u). \tag{2}$$



(a) traditional routing

(b) Plain Network Coding

(c) Compute-and-Forward

Figure 1: Illustration of constraints for the three transmission modes: Communication on the thick edge implies that no useful communication is possible on thin edges.

**No Superposition:** For any $u$ and $t$ it holds that

$$Y_t(u) = \begin{cases} A_t(u-1,u) & \text{if } A_t(u+1,u) = \sigma, \\ A_t(u+1,u) & \text{if } A_t(u-1,u) = \sigma, \\ \tau & \text{if } X_t(u-1) \neq \sigma \text{ and } X_t(u+1) \neq \sigma. \end{cases} \tag{3}$$

**Superposition:** For any $u$ and $t$ it holds that

$$Y_t(u) = A_t(u-1,u) + A_t(u+1,u), \tag{4}$$

where the addition is in $\mathbb{F}_q$, with the additional rules that $X + \sigma = X$ for any symbol $X$, and that $X + \tau = \tau'$ for any $X$ and $\tau$, where $\tau'$ is, like $\tau$, a uniformly distributed random variable from $\mathbb{F}_q$ which is useless to the receiver.

Then, we characterize traditional routing, plain NC and CF as:

$$\begin{array}{rl} \textbf{Traditional Routing:} & \text{neither broadcast nor superposition,} \\ \textbf{Plain Network Coding:} & \text{broadcast, but no superposition,} \\ \textbf{Compute-and-Forward:} & \text{both broadcast and superposition,} \end{array} \tag{5}$$

which can be further specified as:

**Traditional Routing:**

$$\begin{array}{c} \text{If } A_t(u,u+1) \notin \{\tau,\sigma\}, \text{ and } Y_t(u+1) \neq \tau, \text{ then} \\ A_t(u-1 \pm 1, u-1) \in \{\tau,\sigma\}, A_t(u-1,u) = \{\tau,\sigma\}, \\ A_t(u+1,u+1 \pm 1) = \sigma, A_t(u+2,u+2 \pm 1) = \sigma. \end{array} \tag{6}$$

**Plain Network Coding:**

$$\begin{array}{c} \text{If } A_t(u,u+1) \notin \{\tau,\sigma\}, \text{ and } Y_t(u+1) \neq \tau, \text{ then} \\ A_t(u-2,u-1) \in \{\tau,\sigma\}, A_t(u-1,u) \in \{\tau,\sigma\}, \\ A_t(u+1,u+1 \pm 1) = \sigma, A_t(u+2,u+2 \pm 1) = \sigma. \end{array} \tag{7}$$

**Compute-and-Forward:**

$$\begin{array}{c} \text{If } A_t(u,u+1) \notin \{\tau,\sigma\}, \text{ and } Y_t(u+1) \neq \tau, \text{ then} \\ A_t(u-1,u) \in \{\tau,\sigma\}, A_t(u+1,u+1 \pm 1) = \sigma. \end{array} \tag{8}$$

These transmission models are illustrated in Figure 1.

Next, we specify the random access scheduling mechanism. We assume all nodes apply the plain random access approach, i.e., in each time slot each node chooses its state to be "Transmitting" or "Receiving" with probability $a$ and $1-a$ ($a \in [0,1]$ is a fixed constant), respectively. This choice is independent of the state in other time slots and independent of the other nodes. A node can only transmit when it is "Transmitting", and can only receive when it is "Receiving". For traditional routing, we assume each node has equal probability of transmission to either direction.

# 3 The Scheme

Our scheme is based on random linear network coding. The essence of the scheme is that we guarantee that only innovative messages are transmitted. The scheme consists of the following elements:

- We assume that there are $P$ and $Q$ messages ($P, Q \to \infty$) to be transmitted by the left and right source, respectively. These messages are denoted as $\overrightarrow{\mathcal{X}} = \{\overrightarrow{X}(1), \overrightarrow{X}(2), \ldots, \overrightarrow{X}(P)\}$ and $\overleftarrow{\mathcal{X}} = \{\overleftarrow{X}(1), \overleftarrow{X}(2), \ldots, \overleftarrow{X}(Q)\}$, respectively.

- We assume each node keeps three buffers of sufficiently large size, denoted by $R$, $A$ and $B$, respectively. The use of the various buffers will be explained below.

- We define the messages in the buffers of node $m$ as $\mathcal{Y}_m^R = \{Y_m^R(i) : i = 1, 2, \ldots, N_m^R\}$, $\mathcal{Y}_m^A = \{Y_m^A(i) : i = 1, 2, \ldots, N_m^A\}$ and $\mathcal{Y}_m^B = \{Y_m^B(i) : i = 1, 2, \ldots, N_m^B\}$. Since all the messages and all the coefficients are chosen from $\mathbb{F}_q$ and we only do linear coding, we can express the messages as

$$Y_m^R(i) = \sum_{j=1}^{P} \alpha_m^R(i,j)\overrightarrow{X}(j) + \sum_{j=1}^{Q} \beta_m^R(i,j)\overleftarrow{X}(j), \tag{9}$$

$$Y_m^A(i) = \sum_{j=1}^{P} \alpha_m^A(i,j)\overrightarrow{X}(j) + \sum_{j=1}^{Q} \beta_m^A(i,j)\overleftarrow{X}(j), \tag{10}$$

$$Y_m^B(i) = \sum_{j=1}^{P} \alpha_m^B(i,j)\overrightarrow{X}(j) + \sum_{j=1}^{Q} \beta_m^B(i,j)\overleftarrow{X}(j). \tag{11}$$

- Next, we construct a $N_m^R \times P$ matrix $\overrightarrow{H}_m^R$, by setting $\alpha_m^R(i,j)$ as the element in its $i$th row and $j$th column, and a $N_m^R \times Q$ matrix $\overleftarrow{H}_m^R$, by setting $\beta_m^R(i,j)$ as the element in its $i$th row and $j$th column. Then, similarly, we construct $\overrightarrow{H}_m^A$ and $\overleftarrow{H}_m^B$ with $\alpha_m^A(i,j)$ and $\beta_m^B(i,j)$, respectively.

- We assume that node $m$ knows the matrices $\overrightarrow{H}_m^R$ and $\overleftarrow{H}_m^R$, i.e. a node knows which linear combination of messages is being received. This can be guaranteed if we allow the coding coefficients to be communicated without compute-and-forward.

Since each node has at most two neighbours and the topology of the network is fixed, this can be achieved at negligible overhead.

Now, the scheme operates as follows. Initially, we assume that all the buffers in all nodes are empty. In each time slot all nodes perform the following steps:

**Step 1** For each node, all receptions directly enter $R$ (for the sources nodes on both ends of the network, the original messages directly enter $R$).

**Step 2** At the beginning of each time slot, each node updates the matrices $\overrightarrow{H}^R_m$, $\overleftarrow{H}^R_m$, $\overrightarrow{H}^A_m$, and $\overleftarrow{H}^B_m$. Initialize two messages $\overrightarrow{Y} = \overleftarrow{Y} = \sigma$ ($\sigma$ is defined in Section 2).

**Step 3** If $\mathrm{rank}(\overrightarrow{H}^R_m) \leq \mathrm{rank}(\overrightarrow{H}^A_m)$, skip this step. Otherwise, compute a random linear combination of all the $Y^R_m(i)$, denoted as $\overrightarrow{Y} = \sum_{j=1}^{P} \alpha'(j)\overrightarrow{X}(j) + \sum_{j=1}^{Q} \beta'(j)\overleftarrow{X}(j)$. We denote vector $\overrightarrow{V} = \{\alpha'(1), \alpha'(2), \ldots, \alpha'(P)\}$. If the $\overrightarrow{V}$ is not linearly independent of all the rows in $\overrightarrow{H}^A_m$, then it discards $\overrightarrow{Y}$ and regenerates another linear combination $\overrightarrow{Y}$ until $\overrightarrow{V}$ is linearly independent of all rows in $\overrightarrow{H}^A_m$.

**Step 4** Then, each node does the same check for $\overleftarrow{H}^R_m$ and $\overleftarrow{H}^B_m$, and generates $\overleftarrow{Y}$ similarly if $\mathrm{rank}(\overleftarrow{H}^R_m) > \mathrm{rank}(\overleftarrow{H}^B_m)$.

**Step 5** If a node $m$ is "Transmitting" in this time slot, it broadcasts $\overrightarrow{Y} + \overleftarrow{Y}$. If this transmission is successful to the right (the transmitted message is received by its right neighbor, i.e., the right neighbor is at the state "Receiving" in this time slot), then it adds $\overrightarrow{Y}$ to buffer $A$. Symmetrically, $\overleftarrow{Y}$ is added to $B$ if this transmission is successful to the left. However, if the transmission is not successful, then nothing will be added to $A$ and $B$, and the node discards $\overrightarrow{Y}$ and $\overleftarrow{Y}$.

# 4 Performance

In this section we present the key result of the current paper, the maximum achievable throughput of CF on the line network with random access.

**Theorem 1.** *For a line network with random access, the capacity for both directions is $a(1-a)$ with CF. This capacity can be achieved by using the random PLNC scheme introduced in Section 3.*

*Proof.* **Achievability:** We first consider only the session from left to right, assuming that the right source is not transmitting anything. In this case we can interpret the operation of the scheme as follows. Innovative packets are carried through the network over a series of links from left to right. These links are unreliable in the sense that due to random access and half duplex constraints they are not always available. Observe that the model that we described above is exactly the model studied by Lun et al. in [10]. Therefore, it follows directly from [10, Theorem 2] that we can achieve a rate $a(1-a)$ for the left-to-right session. Note, that the use of compute-and-forward does not require any generalizations of the models from [10] in which only broadcast, but not compute-and-forward, is allowed. Even though our underlying model has compute-and-forward, the abstraction described above is that of innovative packets being transmitted over a directed graph.

Next, we include the other session. Observe, that the two sessions can be analyzed independently. More precisely, if both sources are transmitting packets, the flow of innovative packets for each of the sessions can be analyzed by ignoring the other session. Once enough innovative packets are collected at the receiver it can substract the (linear combinations of) packets from the session for which it is the source and decode the packets from the other source. Therefore, we can achieve a rate $a(1-a)$ for both sessions.

**Optimality:** Since the network is symmetric, we focus on the capacity along one direction only. Following the Max-Flow-Min-Cut theorem, the capacity of this network is bounded by the capacity of each edge.

We consider an individual edge. With probability $a(1-a)$, a transmission is made and successfully received. With probability $a^2$, a transmission is made but failed to be received due to the receiver is at the state of "Transmitting". With probability $1-a$, the node is "Receiving", thus no transmission along the edge.

Hence, this channel can be considered as an erasure channel with erasure probability $1-a+a^2$, the capacity of which is $a(1-a)$. As a result, the throughput of our random PLNC scheme is optimal. □

Next, we compare the throughput of random PLNC to that of traditional routing and plain NC presented in [6].

**Theorem 2** ([6], Theorem 6)**.** *For a line network with random access, rates of* $\frac{a(1-a)^2}{2}$ *and* $a(1-a)^2$ *can be achieved by using traditional routing and plain NC, respectively.*

Figure 2 shows the comparison of the rate of either centralized scheduling (CS) and random access (RA) between traditional routing (TR), plain NC (PNC) and CF. Accordingly, the performance of the random PLNC scheme is labeled as "CF, RA". As observed, our random PLNC scheme significantly outperforms the traditional routing and plain NC for the random access scenario. If we allow the probability of transmission $a$ to be adjusted to the modes, then the rates will be maximized in traditional routing and plain NC when $a = 1/3$. The maximum achievable rates are 0.074 and 0.148, respectively. With random PLNC, the maximum rate 0.25 is achieved when $a = 1/2$. Hence, the maximum achievable rates of traditional routing and plain NC are improved by factors of 3.378 and 1.689, respectively. These factors are significantly higher than for the case of centralized scheduling, which has factors of 2 and 1.5 [4].

Figure 3 shows the ratio between the rates of random access and the rates of centralized scheduling for various transmission modes. In other words, it shows the compatibility of these transmission modes with random access. As the figure shows, CF allows the random access scheduling mechanism to utilize the network in a relatively efficient way.

Figure 4 shows the improvement of the throughput of CF over traditional routing and plain NC when fixing the value of $a$. With deterministic centralized scheduling, capacity achieving PLNC schemes have been proposed in [2] [4] on the line network,

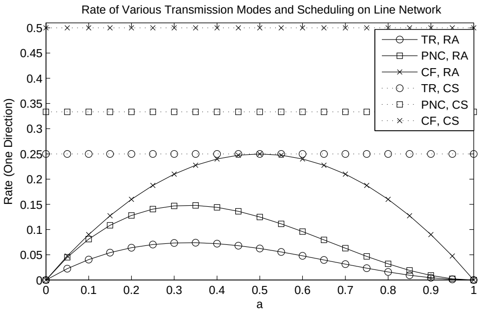


Figure 2: The rates for various scheduling and transmission modes

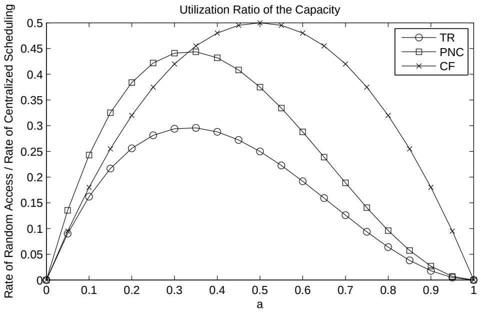


Figure 3: The ratio between the rates of random access and centralized scheduling

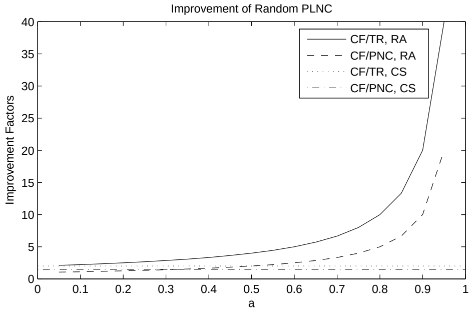


Figure 4: The improvement of CF over TR and PNC

which have improvement factors 2 and 1.5 over traditional routing and plain NC, respectively. However, on the line network with random access, with our random PLNC scheme, improvement factors of $\frac{2}{1-a}$ and $\frac{1}{1-a}$ are obtained, which approach infinity when $a$ approaches 1, as shown in the figure.

# 5 Conclusions and Recommendations

In this paper, we have proposed a random PLNC scheme which can be used in a line network with random access scheduling, and we have proved that this scheme is optimal in this scenario. This result not only shows the feasibility of combining CF and random access, but also indicates that CF can have even greater improvement over traditional routing and plain NC than the improvements offered under centralized deterministic scheduling. This study can be used as a basis for other studies considering CF or PLNC with random access, which might consider more general configurations with less restrictive assumptions, e.g. multiple sessions, incoherent networks, two-dimensional networks, etc. Furthermore, other decentralized scheduling schemes and MAC schemes could be considered as well, especially for some MAC schemes particularly designed for PLNC. One such study is presented in [11].

# Acknowledgment

This work was supported by ERC Starting Grant 259530-ComCom and by NWO Grant 612.001.107.

# References

[1] B. Nazer and M. Gastpar, "Reliable physical layer network coding," *Proc. IEEE*, vol 99, pp. 438-460, 2011.

[2] S. Zhang, S. Kiew, and P. Lam, "Hot topic: Physical-layer network coding," in *Proc. 12th Annu. Int. Conf. Mobile Computing and Networking*, pp. 358-365, 2006.

[3] S. Katti, S. Gollakota and D. Katabi, "Embracing wireless interference: Analog network coding," *Proc. ACM SIGCOMM*, pp. 397-408, 2007.

[4] J. Goseling, M. Gastpar, and J. H. Weber, "Line and lattice networks under deterministic interference models," *IEEE Trans. Inf. Theory*, vol. 57, pp. 3080-3099, 2011.

[5] Z. Ren, J. Goseling, J. H. Weber and M. Gastpar, "Compute-and-forward: multiple bi-directional sessions on the line network," to be published in *Proc. ISIT*, 2013.

[6] Y. E. Sagduyu, A. Ephremides, "Cross-layer optimization of MAC and network coding in wireless queueing tandem networks," in *IEEE Trans. Inf. Theory*, vol. 54, pp 554-571, 2008.

[7] D. Umehara, T. Hirano, S. Denno, M. Morikura and T. Sugiyama,"Wireless network coding in slotted aloha with two-hop unbalanced traffic," *Selected Areas in Communications, IEEE Journal*, pp. 647-661, 2009.

[8] J. Goseling, M. Gastpar and J. H. Weber, "Random access with physical-layer network coding," *Information Theory and Application Workshop (ITA)*, pp. 1-7, 2013.

[9] P. Pakzad, C. Fragouli, and A. Shokrollahi, "Coding schemes for line networks," *Proc. of ISIT*, pp. 1853-1857, 2005.

[10] D. S. Lun, M. Médard, R. Koetter and M. Effros, "On coding for reliable communication over packet networks," *Phy. Comm.*, vol. 1, pp 3-20, 2008.

[11] S. Wang, Q. Song, X. Wang and A. Jamalipour, "Distributed MAC protocol supporting physical-layer network coding," *Mobile Comp., IEEE Trans.*, vol. 12, pp. 1023-36, 2012.