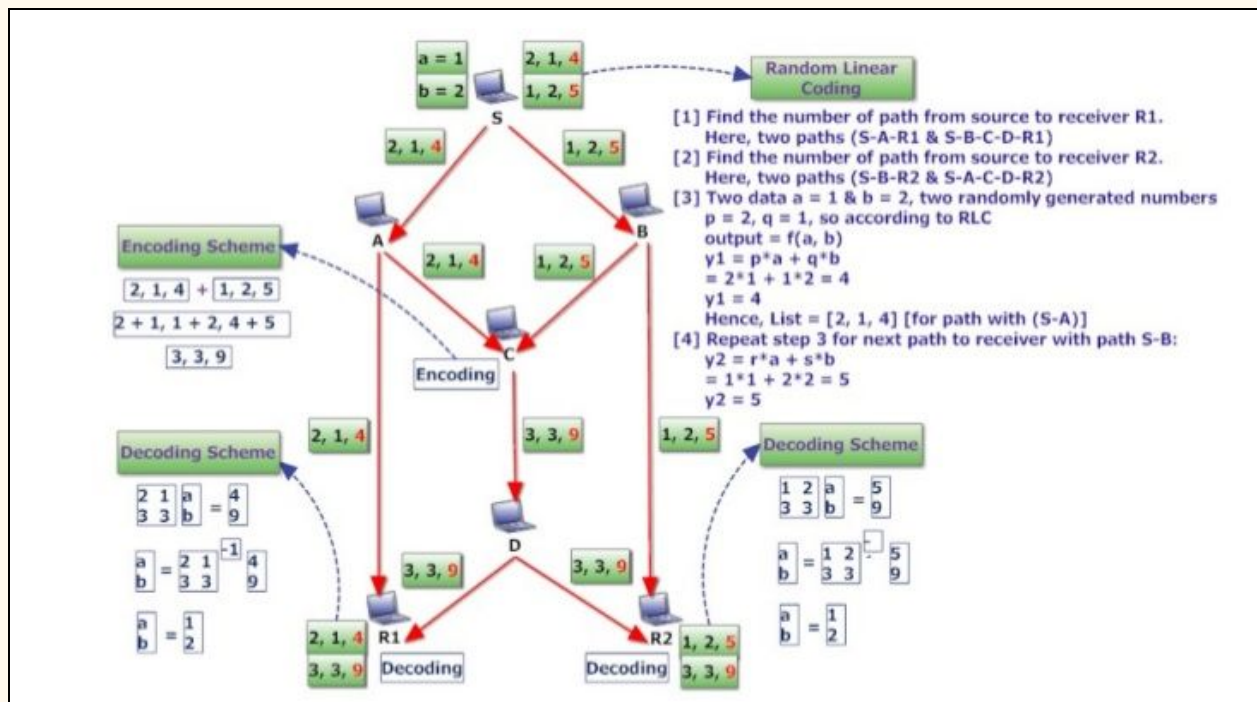


# THE ONE NETWORK CODING IMPLEMENTATION

Diogo Soares



## INTRODUÇÃO

Alguns tipos de redes como as redes MANETs, redes oportunistas, redes D2D viabilizam incrementar o alcance na transmissão de mensagens sem aumentar a carga de dados sob uma autoridade central como antenas ou servidores em nuvem. Para isso, durante a transmissão de mensagens (de modo *unicast* ou *multicast*), os nós podem atuar como servidores *locais*, armazenando, de modo intermediário, mensagens que serão retransmitidas até alcançar o(s) nó(s) destinatário(s) como visto na Figura 1.

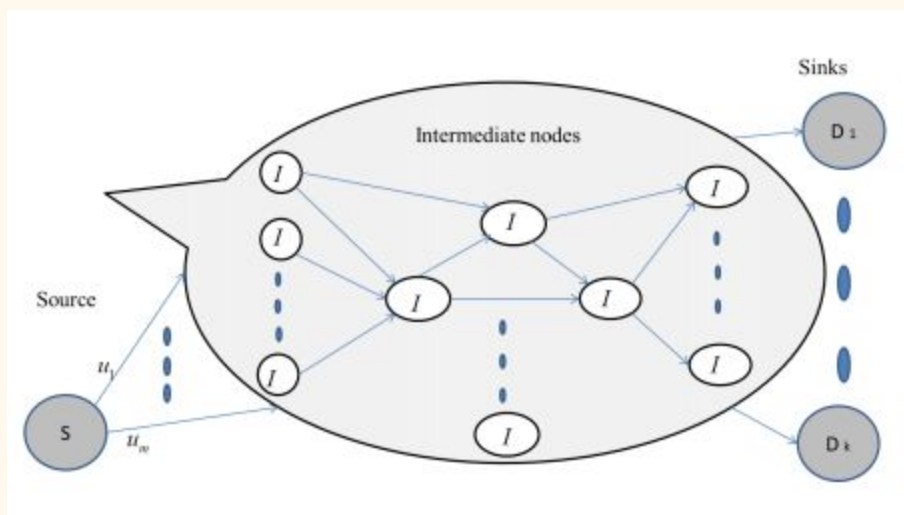


Figura 1. Transmissão de dados em redes D2D.

Este trabalho visa a compreensão do conceito de *network coding* e como sua aplicação pode ser útil e necessária em cenários D2D.

## Network Coding

Técnica utilizada para múltiplos pacotes em apenas um através de técnicas matemáticas com o objetivo de propagar mais informação sem aumentar a quantidade de informações propagadas na rede. Desse modo essa técnica é aplicada com o objetivo de melhorar o *throughput* da rede, eficiência, latência, escalabilidade, e até para prevenção de ataques e *eavesdropping*.

## Algoritmos de Network Coding

Existem diversas técnicas<sup>1</sup> para realização de *network coding*. Redes em que os nós se comunicam pelo mesmo meio de transmissão criam várias oportunidades para que eles escutem pacotes trafegando na rede e possam auxiliar na comunicação ao atuarem de modo oportunístico, carregando a mensagem na rede.

### 1. XOR

A técnica de disjunção exclusiva (XOR) possui as vantagens do baixo processamento computacional tanto para codificação como decodificação.

<sup>1</sup> <http://www2.ece.rochester.edu/projects/wcng/meetings/PerilloNetworkCoding.pdf>

De modo geral, a técnica XOR funciona da seguinte forma: para uma sequência  $m$  de mensagens com  $n$  bits cada, uma combinação linear é gerada a partir da operação XOR bit a bit entre as  $m$  mensagens. Uma exemplificação dessa técnica é vista na Figura 2.

$$\begin{array}{r}
 \text{XOR} \left\{ \begin{array}{l} a_1 \ a_2 \ a_3 \ \dots \ a_n \\ b_1 \ b_2 \ b_3 \ \dots \ b_n \end{array} \right. \\
 \hline
 \text{XOR} \left\{ \begin{array}{l} (a_1 \text{ xor } b_1) \ \dots \ (a_n \text{ xor } b_n) \\ \vdots \\ z_1 \ z_2 \ z_3 \ \dots \ z_n \end{array} \right. \\
 \hline
 (a_1 \text{ xor } b_1 \text{ xor } \dots \text{ xor } z_1) \ \dots \ (a_n \text{ xor } b_n \text{ xor } \dots \text{ xor } z_n)
 \end{array}$$

Figura 2. Operação XOR.

## 2. Linear Network Coding

A técnica de *linear network coding* (LNC) usa uma abordagem diferenciada e ao mesmo tempo semelhante ao XOR. LNC utiliza um conjunto de informações adicionais acrescentadas aos pacotes transmitidos e é assumido que as operações executadas nos *hosts* sejam de ordem algorítmica linear. Adicionalmente, informações extras são adicionados ao cabeçalho da mensagem retransmitida com esta codificação. Para efeitos de informação, tais informações adicionais são os coeficientes necessários para realização desta técnica como explicado mais adiante.

Formalmente, no LNC, os símbolos utilizados na codificação do pacote são retirado de um campo finito  $F_q$  e um pacote (ou os dados do pacote) é um vetor sobre de símbolos sob  $F_q$ . Assim, o alfabeto do pacote  $\beta$  é um espaço de linhas do vetor sob  $F_q$ . O menor campo finito possível é chamado  $GF(2)$ , também conhecido como campo de galois, que possui apenas os elementos 0 e 1<sup>23</sup>. Um exemplo das funções aritméticas básicas de adição e multiplicação para  $GF(2)$  podem ser vistas nas tabela

+	0	1
---	---	---

<sup>2</sup> <http://web.mit.edu/medard/www/allerton3.pdf>

<sup>3</sup> <https://crypto.stackexchange.com/questions/2700/galois-fields-in-cryptography>

0	0	1
1	1	0

Tabela 1. Adição em  $GF(2)$ .

+	0	1
0	0	0
1	0	1

Tabela 2. Multiplicação em  $GF(2)$ .

Para melhor compreensão do LNC, suponha o exemplo de transmissão da Figura 3. *Host* 1 cria mensagens  $x_1$  e  $x_2$  e encaminha para *hosts* 6 e 7 através de alguns *hosts* intermediários (2, 3, 4 e 5). Mensagens são retransmitidas como combinação linear entre  $m = 2$  pacotes usando 2 coeficientes para gerar uma mensagem única, esta que é retransmitida, adicionando-se os coeficientes ao cabeçalho da mensagem. No *host* 4 as mensagens  $y_1$  e  $y_2$  (combinações lineares de  $x_1$  e  $x_2$ ), este realiza uma nova combinação linear (usando coeficientes  $\beta_1$  e  $\beta_2$ ) para gerar mensagem  $Y$ . Esta é retransmitida normalmente para *host* 5. Quando  $Y$  chega aos *hosts* 6 e 7, o processo de decodificação é feito através da operação linear de multiplicação entre a matriz inversa de coeficientes recebidos com o vetor de mensagens codificadas recebidas,  $Y$  para ambos os *hosts*, além de  $y_1$  para o *host* 6 e  $y_2$  para o *host* 7, assim decodificando as mensagens originais  $x_1$  e  $x_2$ .

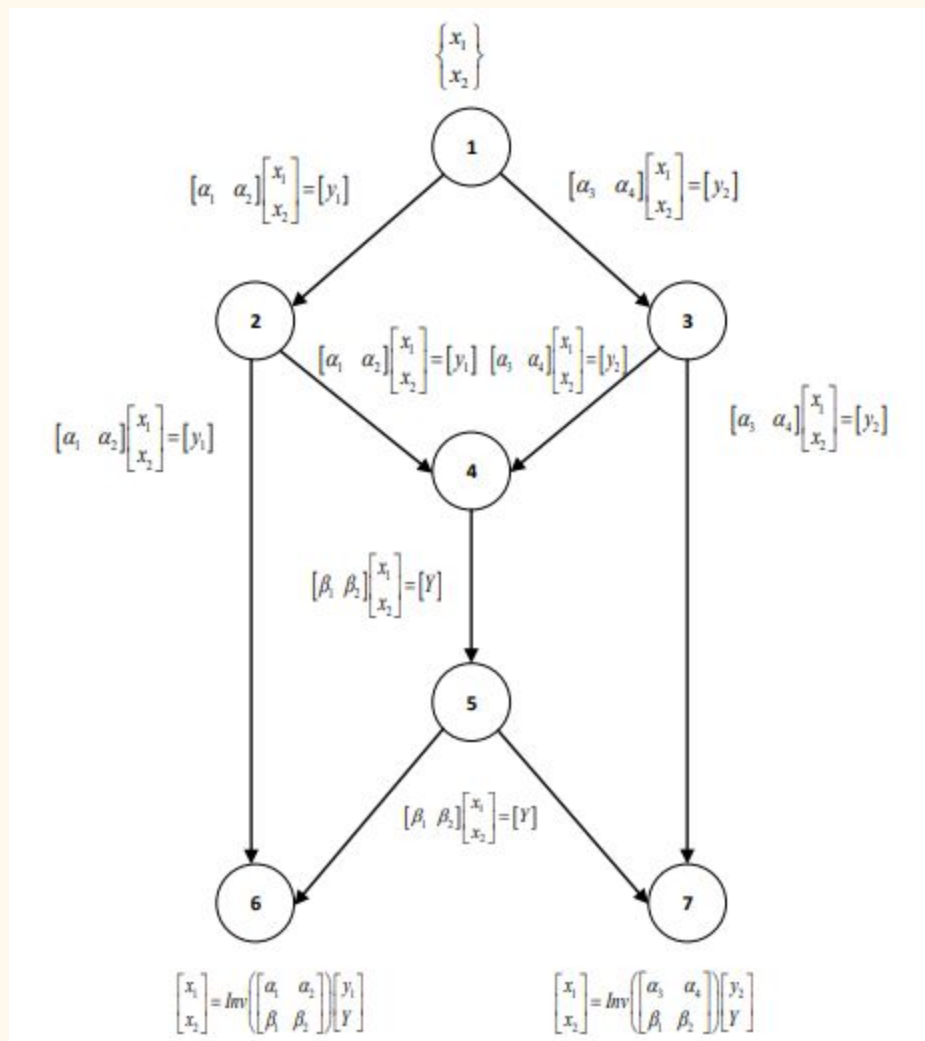


Figura 3. Exemplo da topologia borboleta para LNC.

## Desafios de Network Coding

Alguns desafios relacionados ao *network coding*:

### 1. XOR

Embora seja uma técnica de baixo poder computacional para resolução, o agrupamento contínuo de múltiplas mensagens podem causar um grande *delay* na fase de decodificação devido a quantidade mínima de informação necessária para decodificação.

## 2. *Linear Network Coding*

Variedade de parâmetros a serem considerados na fase de implementação pode afetar desempenho geral da rede e é um campo de estudos em aberto em redes. Número de pacotes usados na codificação, por exemplo, pode afetar o *throughput* e o *delay* segundo trabalhos anteriores<sup>4</sup>. A escolha do tamanho do campo finito utilizado também pode gerar complexidade algorítmica durante a decodificação além de overhead na mensagem.

# IMPLEMENTAÇÃO

Para acesso ao código do modelo de simulação utilizado, acesse: [the-one-forked](#).

## Rodando o The ONE

Para rodar o The ONE (linux + Java JDK):

```
$ chmod u+x *.sh; ./compile.sh
```

```
$ ./one.sh
```

Informações adicionais sobre execução e configuração veja o arquivo *README.txt* na pasta **the-one**.

## Parametrização da Simulação

A parametrização da simulação é feita no arquivo *defaultl\_settings.txt*. Para edição de um novo modelo de simulação e execução faça:

```
$ vim ncParams.txt
```

```
$ ./one.sh ncParams.txt
```

No arquivo de parametrização, considere os principais parâmetros de simulação:

- **Tempo de simulação em segundos:** “Scenario.endTime = 60”.

---

<sup>4</sup> <http://www.mit.edu/~medard/mpapers/xors.pdf>

- **Configuração do grafo de contato entre os *hosts*:** Defina “Events1.class = ExternalEventsQueue” e o arquivo representando o grafo de contato “Events1.filePath = traces/traceContato.txt”. O formato desse arquivo é definido pelo padrão:

TIME	CONN	FROM_HOST	TO_HOST	TYPE
------	------	-----------	---------	------

**TYPE** é definido por {*up*, *down*}, no qual *up* representa uma conexão no grafo e *down*, sua desconexão.

- **Configuração de geração de mensagens:** Defina “Events1.class = ExternalEventsQueue” e o arquivo representando a geração de mensagens “Events2.filePath = traces/traceGeracaoMensagens.txt”. O formato desse arquivo é definido pelo padrão:

TIME	TYPE	MESSAGE_ID	FROM_HOST	TO_HOST	SIZE
------	------	------------	-----------	---------	------

## Arquitetura do Sistema

Nesta seção é descrito a arquitetura do sistema que vai ser implementada no simulador.

### 3. Pacote de Mensagem

Cada mensagem é criada com uma variável de dados. Aqui ressaltamos que a implementação em alto nível não representa com total precisão a estrutura de um *data packet*. Aqui o intuito é apenas simular o comportamento e codificação.

Para isso foi implementado em **src/core/Message.java** a geração de um número aleatório representando os dados, no intervalo de  $[0, 1 \ll 8 - 1]$ . Este dado que precisa ser codificado através do *network coding*. Neste trabalho apenas o dado é codificado.

Através da classe **src/gui/InfoPanel.java** na função **setMessageInfo()** informações sobre a mensagem (codificação do dado, se a mensagem é atômica ou uma codificação entre *a* e *b*). Um exemplo de alteração pode ser visto na Figura 4.

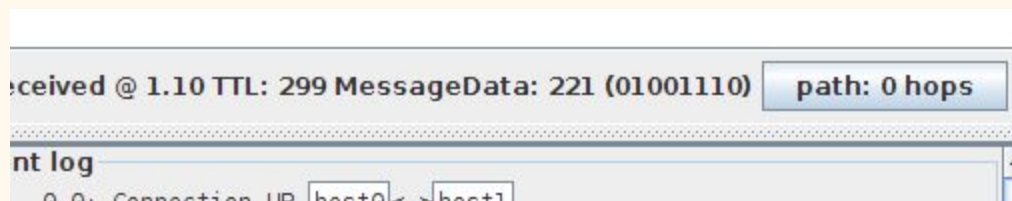


Figura 4. Dado gerado na mensagem.

Informação sobre o fato da mensagem ser atômica ou codificada precisa ser adicionado ao código fonte.

#### 4. Grafo de Contato entre *Hosts*

O grafo de contato entre *hosts* foi dividido em 3 cenários, no qual cada cenário representa uma situação a ser avaliada usando *network coding* e não usando a técnica. Abaixo são apresentados os detalhes de cada um dos 3 cenários:

##### 1. Cenário 1

O cenário 1 consiste de um nó fonte **S** enviando uma mensagem para um nó destino **D**. Essa mensagem é dividida em 5 pacotes e todos precisam chegar em **D**. Para este cenário foi criado um modelo de simulação que pode ser visto na pasta **the-one/traces/\*Cenario1.txt**. Os parametros de simulação podem ser vistos em **the-one/cenario1.txt** no qual é configurado o nó fonte **S**, nó destino **D** e 3 hosts intermediários.

Execute o cenário 1 e verifique a latência obtida. Encontre a latência usando *network coding*.

##### 2. Cenário 2

De modo similar ao cenário 1, no cenário 2 há 2 hosts fontes **S'** e **S''**, 3 hosts intermediários e um nó destino **D**.

##### 3. Cenário 3

O cenário 3 representa um cenário no qual há 2 hosts fontes **S'** e **S''**, 3 hosts intermediários e 2 nós destinos **D'** e **D''**.

### Links de referência

[The ONE - Link oficial](#)

[Cautious View on Network Coding - From Theory to Practice](#)

[Kodo: An Open and Research Oriented Network Coding Library](#)

[Delay-Tolerant Networks with Network Coding: How Well Can We Simulate Real Devices?](#)

[Network Coding Algorithms and Applications](#)

[Simulation of Random Linear Network Coding in Ad-Hoc Networks](#)



[Implementation of Linear Network Coding Over a Flexible Emulator](#)

[Esquemas de Segurança contra Ataques de Poluição em Codificação de Rede sobre Redes sem Fios](#)

[Decoding Algorithms for Random Linear Network Codes](#)

[The ONE for beginners](#)

[The ONE Knowledge Base - Infos extraídos do fórum oficial](#)

[Site do Sahil Gupta com várias documentações](#)

[Métricas do The ONE](#)

[Source and destination of messages on The ONE](#)

[Controle externo de mensagens no The ONE](#)

[The ONE - Advanced Features](#)

[The ONE - External Resources](#)

[DTN blog](#)

[CodePLC: um Protocolo de Network Coding MAC para Power Line Communication](#)

[Decoding Delay Performance of Random Linear Network Coding for Broadcast](#)

[Network Coding for Delay Constrained Wireless Systems with Feedback](#)

[Different technologies in vehicular networks : multihoming and network coding](#)

[Análise de métricas de desempenho para uma rede AD HOC Slotted - Aloha de dois saltos com Network Coding](#)