Cooperation and Coding

Monica Navarro

Centre Tecnològic de Telecomunicacions de Catalunya (CTTC)

Wireless Networks: From Energy Harvesting to Information Processing

European School of Antennas/WIPE-COST ACTION IC1301

9 – 13 Nov. 2015, Castelldefels, Spain

Centre
Tecnològic
de Telecomunicacions
de Catalunya

Outline

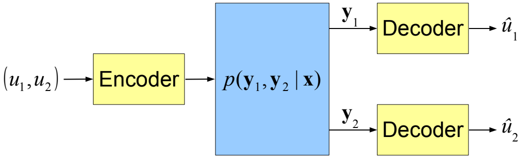
1. Multi-user Information Theory
   - Capacity regions of broadcast and multiple-access channel
   - Uplink-downlink duality
2. Cooperative Schemes
   - The relay channel: capacity
   - Protocols
   - Coded Cooperation
     - Outage probability
     - Implementation example
3. Coding at upper layers
   - Basics of Network Coding (NC)
   - Rateless codes

Multi-User Information Theory

Multi-User Channels

- Multiple users add another dimension
  - Multi-user diversity, scheduling, . . .
- Capacity cannot be characterized by a single number
  - Define a K-dimensional capacity region
  - Several optimization criteria are possible
  - "Fairness" between users
- Different power constraints possible
  - e.g. in uplink K power constraints, in downlink only one

The Broadcast Channel



- Capacity region for general case not known
- Broadcast channel is memoryless iff $p(\mathbf{y}_1, \mathbf{y}_2|\mathbf{x}) = \prod_{i=1}^{n} p(y_{1i}, y_{2i}|x_i)$
- The broadcast channel is said to be degraded iff

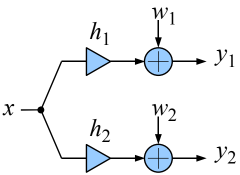
$$p(y_1, y_2|x) = p(y_1|x)p(y_2|y_1)$$

  i.e. $p(y_2|y_1, x) = p(y_2|y_1)$ and $x \to y_1 \to y_2$ form a Markov chain.
- The capacity region for the degraded BC is known

The Gaussian Broadcast Channel

- Definitions
  - $x, y \in \mathbb{C}$, AWGN, $w_k \sim \mathcal{CN}(0, N_k)$, power gains $|h_k|^2$, power constraint $\sum_{k=1}^{K} P_k = P$
  - channel gain to noise ratio (CNR): $T_k \triangleq \frac{|h_k|^2}{N_k}$
  - for simplicity, consider two-user channel
  - $T_1 \geq T_2$, i.e. user 1 has better channel
  - CNRs of users can be ordered
    $\Rightarrow$ *degraded* broadcast channel
  - Rate region is defined as the union of all achievable rates (*independent data*)

$$\mathcal{C}_{\mathrm{BC}} = \bigcup (R_1, R_2)$$



The Gaussian Broadcast Channel

- Corner points of rate region: all resources (bandwidth, time, power) are allocated to one user: $R_k^{(\mathrm{c})} = \mathrm{ld}(1 + T_k P)$
- Equal power time division: for $\sum_k \alpha_k = 1,\ \alpha_k \geq 0$, we obtain a straight line between corner points

$$R_k = \alpha_k \mathrm{ld}(1 + T_k P)$$

- Variable-power time division: for $\sum_k \alpha_k P_k = P$

$$R_k = \alpha_k \mathrm{ld}(1 + T_k P_k)$$

- Frequency division: with $\sum_k \beta_k = 1,\ \beta_k \geq 0$

$$R_k = \alpha_k \mathrm{ld}\left(1 + T_k \frac{\beta_k P}{\alpha_k}\right)$$

  - by setting $P_k = \frac{\beta_k}{\alpha_k} P$, we see that the last two regions are identical

The Gaussian Broadcast Channel

- CDMA with non-orthogonal spreading codes, spreading gain G, without interference cancellation

$$R_1 = \frac{1}{G}\mathrm{ld}\left(1 + \frac{\alpha_1 PG}{1/T_1 + \alpha_2 P}\right)$$
$$R_2 = \frac{1}{G}\mathrm{ld}\left(1 + \frac{\alpha_2 PG}{1/T_2 + \alpha_1 P}\right) \quad (1)$$

- BC region: superposition coding with successive interference cancellation:

$$R_1 = \mathrm{ld}\left(1 + T_1\alpha_1 P\right)$$
$$R_2 = \mathrm{ld}\left(1 + \frac{\alpha_2 P}{1/T_2 + \alpha_1 P}\right) \quad (2)$$

The Gaussian Broadcast Channel

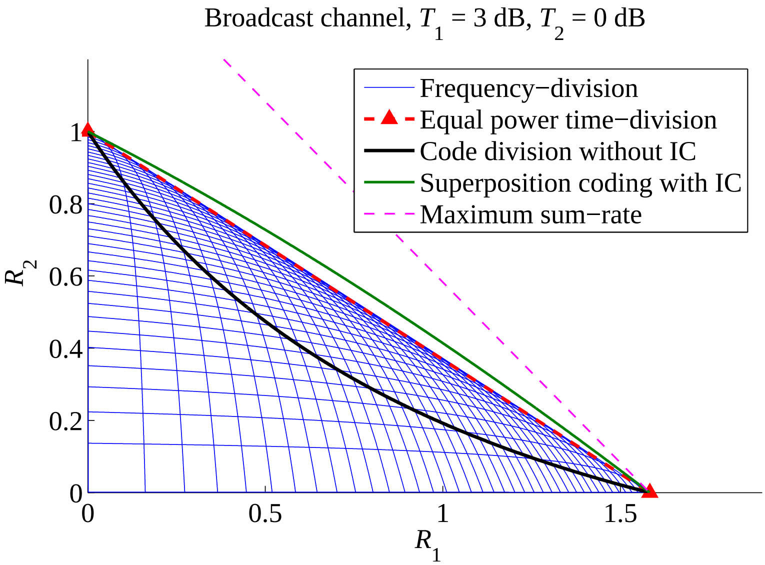
- Maximum sum rate
  - simple figure of merit for multi-user system

$$R_{\mathrm{sum}} = \max_{\mathbf{R} \in \mathcal{C}_{\mathrm{BC}}} \sum_{k=1}^{K} R_k = \mathrm{ld}\left(1 + P \max_k T_k\right) \tag{3}$$

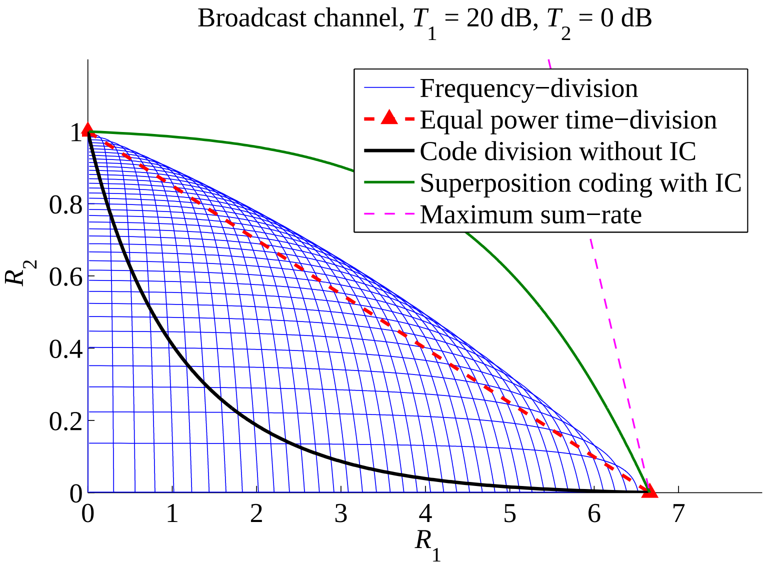
  - is achieved at boundary point of best user
    $\Rightarrow$ BC reduces to single-user system
- Maximum symmetric rate
  - all users obtain the same rate

$$R_{\mathrm{sym}} = \max_{\mathbf{R} \in \mathcal{C}_{\mathrm{BC}}, R = R_k} R \tag{4}$$

The Gaussian Broadcast Channel



The Gaussian Broadcast Channel



The Gaussian Broadcast Channel

- Common data
  - Common data is sent to all users (broadcast)
  - Users receive data of all other users with worse channel
    $\Rightarrow$ include common data in the stream for user with worst channel
  - Rate region with common data, sent to both users at rate R_0:

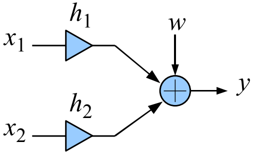
$$
\begin{aligned}
R_0 &\leq \mathrm{ld}\left(1 + \frac{\alpha_2 P}{1/T_2 + \alpha_1 P}\right) \\
R_1 &\leq \mathrm{ld}\left(1 + \alpha_1 P T_1\right) \\
R_2 &\leq \mathrm{ld}\left(1 + \frac{\alpha_2 P}{1/T_2 + \alpha_1 P}\right) - R_0
\end{aligned}
\tag{5}
$$

The Multiple Access Channel

- Channel gain to noise ratio (CNR): $T_k = \frac{|h_k|^2}{N_0}$
- Power constraint per user: $\mathbb{E}[|x_k|^2] \leq P_k, \; k = 1, \ldots, K$

- The capacity region for two users is a pentagon:

$$\begin{aligned} R_1 &\leq I(X_1; Y|X_2) \\ R_2 &\leq I(X_2; Y|X_1) \\ R_1 + R_2 &\leq I(X_1 X_2|Y) \end{aligned} \quad (6)$$

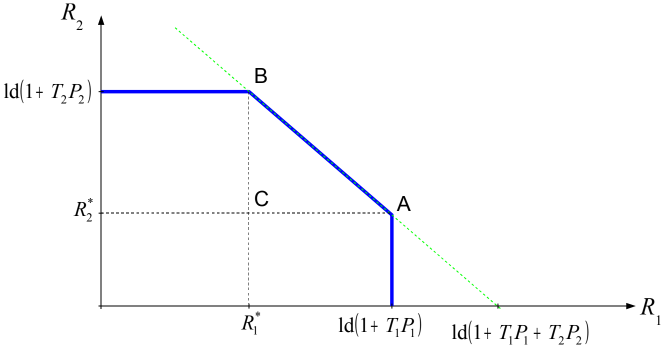


- For Gaussian MAC:

$$\begin{aligned} R_1 &\leq \mathrm{ld}\,(1 + P_1 T_1) \\ R_2 &\leq \mathrm{ld}\,(1 + P_2 T_2) \\ R_1 + R_2 &\leq \mathrm{ld}\,(1 + P_1 T_1 + P_2 T_2) \end{aligned} \quad (7)$$

The Gaussian Multiple Access Channel

- Point A:
  $R_2^* = \mathrm{ld}\left(1 + T_1P_1 + T_2P_2\right) - \mathrm{ld}\left(1 + T_1P_1\right) = \mathrm{ld}\left(1 + \frac{T_2P_2}{1+T_1P_1}\right)$
  decode user 2, treating signal from user 1 as interference, subtract signal, then decode user 1 (successive decoding)



The Gaussian Multiple Access Channel

- The capacity region for K users is

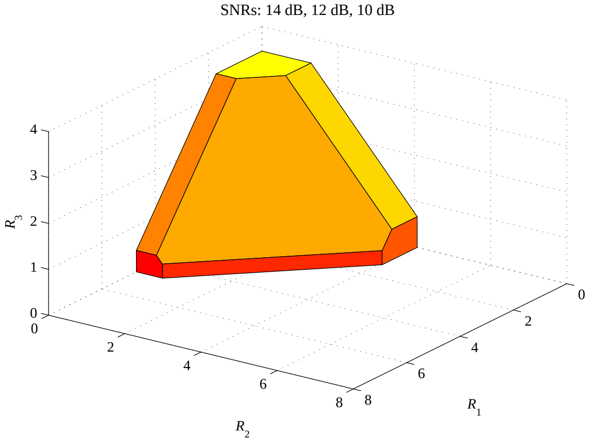
$$\mathcal{C}_{\mathrm{MAC}} = \left\{ \mathbf{R} : \sum_{k \in \mathcal{S}} R_k \leq \mathrm{ld} \left(1 + \sum_{k \in \mathcal{S}} P_k T_k \right), \; \forall \mathcal{S} \subset \{1, 2, \ldots, K\} \right\} \tag{8}$$

- The MAC region has $K!$ vertices in the positive orthant, all achievable with successive decoding with one the $K!$ orderings.
- The set of users $\{1, 2, \ldots, K\}$ has $2^K - 1$ non-empty subsets, i.e. there are $2^K - 1$ conditions on $\mathbf{R}$
- The sum rate is

$$R_{\mathrm{sum}} = \mathrm{ld} \left(1 + \sum_{k=1}^{K} P_k T_k \right) \tag{9}$$

The Gaussian Multiple Access Channel: 3 users

- Capacity region is defined by
  - $2^K - 1 = 7$ inequalities
  - $3! = 6$ vertices in $\mathbb{R}^3_+$ (not counting the ones on $x_i = 0$)



The Gaussian Multiple Access Channel

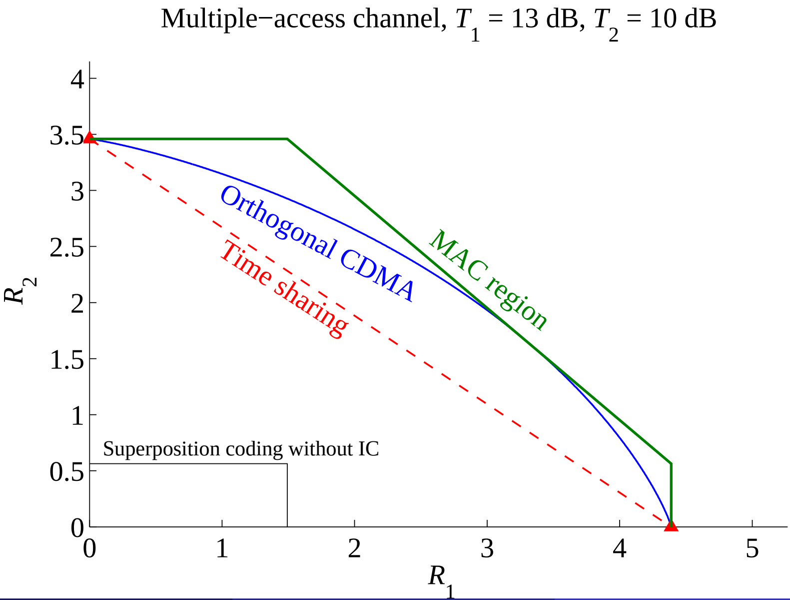
- Suboptimum multiple-access schemes
  - Superposition coding without interference cancelation (differs only at receiver from optimum scheme): The transmit signal of each user appears as noise to all others.

$$R_k = \mathrm{ld}\left(1 + \frac{T_k P_k}{1 + \sum_{\ell \neq k} T_\ell P_\ell}\right) \tag{10}$$

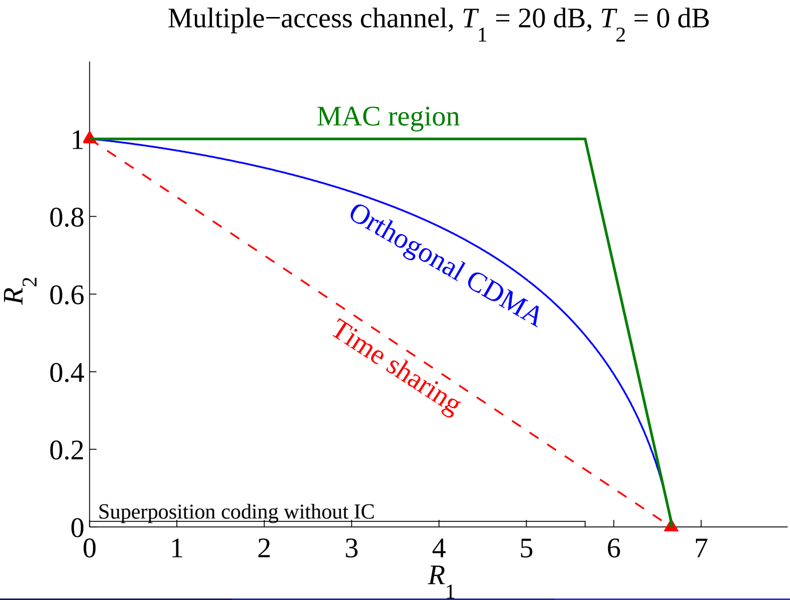
  - Orthogonal CDMA (including frequency and time division)
    Allocate α_k of available bandwidth (or time) to user k, received noise power is then $\alpha_k N_0$

$$R_k = \alpha_k \mathrm{ld}\left(1 + \frac{P_k T_k}{\alpha_k}\right), \quad \sum_{k=1}^{K} \alpha_k = 1,\ \alpha_k \geq 0 \tag{11}$$

The Gaussian Multiple Access Channel



The Gaussian Multiple Access Channel



Sum Rate of Multiple Access Channel

Comparison for equal SNRs: $T_1 P_1 = T_2 P_2 = \cdots = T_K P_K = \gamma$

1. Sum rate for optimal scheme (superposition coding with IC):

$$R_{\mathrm{sum}} = \mathrm{ld}\,(1 + K\gamma)$$

   - rate grows without limit with number of users

2. Sum rate for superposition coding without IC

$$R_{\mathrm{sum}} = K \cdot \mathrm{ld}\left(1 + \frac{\gamma}{1 + (K-1)\gamma}\right) \rightarrow \mathrm{ld}(e) = \frac{1}{\ln 2} = 1.442$$

   for $K \rightarrow \infty$

   - is interference-limited

Broadcast vs. Multiple Access Channel

- **Common:** Optimum scheme is superposition coding with successive interference cancellation

MAC, uplink

- $K!$ decoding orderings, all achieve optimum sum rate

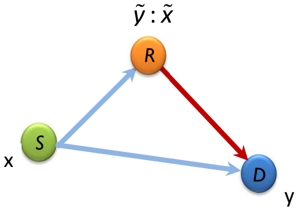
BC, downlink

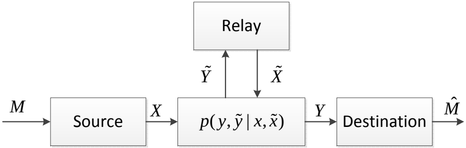
- always decode weakest user first
- optimum sum rate is achieved by transmitting only to strongest user

Cooperative Schemes

The Relay Channel

- Relay channels are known to provide higher capacity than point to point channels
- The capacity of the relay channel is still unknown
- The best known upper bound for the general relay channel is the cut-set bound
- Known capacity for the degraded relay channel





1. The relay channel is defined by the input and output alphabets $\mathbb{X}, \tilde{\mathbb{X}}, \mathbb{Y}, \tilde{\mathbb{Y}}$ and a collection of pmfs $p(y, \tilde{y}|x, \tilde{x})$, one for each $(x, \tilde{x}) \in \mathbb{X} \times \tilde{\mathbb{X}}$
2. An $(n, 2^{nR})$ code for a relay channel consists of
   - message set $\mathcal{U} = \{1, 2, \ldots, 2^{nR}\}$
   - encoding function $X : \mathcal{U} \to \mathbb{X}^n$
   - relay functions $\tilde{x}_i = f_i(\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_{i-1})$
   - decoding function $g : \ Y^n \to U$

- The channel is memoryless: $y_i, \tilde{y}_i$ depend on previously transmitted symbols only via $x_i, \tilde{x}_i$
- Encoding in the relay is causal

The Relay Channel

- The joint distribution factors as

$$p(u, \mathbf{x}, \tilde{\mathbf{x}}, \mathbf{y}, \tilde{\mathbf{y}}) = p(u) \cdot \prod_{i=1}^{n} p(x_i|u) \cdot p(\tilde{x}_i|\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_{i-1}) \cdot p(y_i, \tilde{y}_i|x_i, \tilde{x}_i)$$

Theorem

For any relay channel $\left(\mathbb{X} \times \tilde{\mathbb{X}}, p(y, \tilde{y}|x, \tilde{x}), \mathbb{Y}, \tilde{\mathbb{Y}}\right)$, *the capacity is bounded above by*

$$C \leq \sup_{p(x,\tilde{x})} \min \left\{ I(X, \tilde{X}; Y),\ I(X; Y, \tilde{Y}|\tilde{X}) \right\} \tag{12}$$

Proof by max-flow min-cut theorem

The Relay Channel

Definition

A relay channel is degraded iff

$$p(y, \tilde{y}|x, \tilde{x}) = p(\tilde{y}|x, \tilde{x}) \cdot p(y|\tilde{y}, \tilde{x})$$

i.e. $p(y|x, \tilde{x}, \tilde{y}) = p(y|\tilde{x}, \tilde{y})$

Theorem

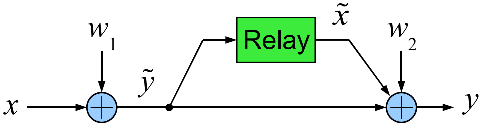
The capacity of the degraded relay channel is

$$C = \sup_{p(x,\tilde{x})} \min \left\{ I(X, \tilde{X}; Y),\ I(X; \tilde{Y}|\tilde{X}) \right\} \tag{13}$$

The Degraded Gaussian Relay Channel

- We consider the *physically degraded* Gaussian relay channel, in which y depends on x only via $\tilde{x}, \tilde{y}$.
- The capacity of the general Gaussian relay channel is not known.

$$\begin{array}{lclcl} \tilde{y} = x + w_1, & & w_1 \sim \mathcal{N}(0, N_1) & & \frac{1}{n}\sum_{i=1}^{n} x_i^2 \leq P_1 \\ & \text{where} & & \text{and} & \\ y = \tilde{y} + \tilde{x} + w_2, & & w_2 \sim \mathcal{N}(0, N_2) & & \frac{1}{n}\sum_{i=1}^{n} \tilde{x}_i^2 \leq P_2 \end{array} \qquad (14)$$



The Degraded Gaussian Relay Channel

Theorem

The capacity of the degraded Gaussian relay channel is

$$C = \max_{0 \leq \alpha \leq 1} \min \left\{ C_{\mathrm{a}} \left(\frac{P_1 + P_2 + 2\sqrt{(1-\alpha)P_1 P_2}}{N_1 + N_2} \right), C_{\mathrm{a}} \left(\frac{\alpha P_1}{N_1} \right) \right\} \quad (15)$$

where $C_{\mathrm{a}}(x) \triangleq \frac{1}{2}\mathrm{ld}(1+x)$.

The Degraded Gaussian Relay Channel

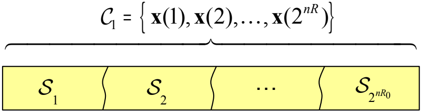
1. Case $\frac{P_1}{N_1} \leq \frac{P_2}{N_2}$: Relay is "closer" to receiver
   - The capacity is determined by the source-relay link as $C_{\mathrm{a}}(P_1/N_1)$ with $\alpha = 1$. Channel appears to be noise free after relay.
   - The rate is increased from $C_{\mathrm{a}}\left(\frac{P_1}{N_1+N_2}\right)$ to $C_{\mathrm{a}}\left(\frac{P_1}{N_1}\right)$.
2. Case $\frac{P_1}{N_1} > \frac{P_2}{N_2}$: Relay is "closer" to sender
   - Capacity is $C_{\mathrm{a}}\left(\frac{\alpha^* P_1}{N_1}\right)$, where α^* such that

$$C_{\mathrm{a}}\left(\frac{P_1 + P_2 + 2\sqrt{(1-\alpha^*)P_1 P_2}}{N_1 + N_2}\right) = C_{\mathrm{a}}\left(\frac{\alpha^* P_1}{N_1}\right)$$

The Degraded Gaussian Relay Channel

Concept for achieving the capacity of the degraded Gaussian relay channel: *Block Markov coding*

- Define two codebooks with rates R and $R_0 < R$.
- First codebook $\mathcal{C}_1 = \left\{\mathbf{x}(u), u = 1, \ldots, 2^{nR}\right\}$, power αP_1
  partition this codebook into 2^{nR_0} cells of equal size



- Second codebook: $\mathcal{C}_2 = \left\{\tilde{\mathbf{x}}(s), s = 1, \ldots, 2^{nR_0}\right\}$, power $(1-\alpha)P_1$

The Degraded Gaussian Relay Channel

- Transmission is organized blockwise: in block i, the sender transmits message u_i, the relay decodes the message and supports the sender in the next block.

Message	u_1	u_2	u_3		
Sender		$\tilde{\mathbf{x}}(s_2)$	$\tilde{\mathbf{x}}(s_3)$	$\cdots$	$(1-\alpha)P_1$
	$\mathbf{x}(u_1)$	$\mathbf{x}(u_2)$	$\mathbf{x}(u_3)$	$\cdots$	αP_1
Relay		$\tilde{\mathbf{x}}(s_2)$	$\tilde{\mathbf{x}}(s_3)$	$\cdots$	P_2

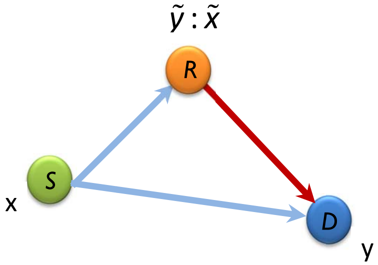
Protocol Strategies

1. Decode-and-Forward (DF)
   - the relay decodes the message transmitted by the source
   - the source uses block Markov encoding
   - In the next block, the relay and source transmit the message to the destination
2. Compress-and-Forward(CF)
   - the relay compresses received symbol (does not decode), transmits to destination
   - The destination uses the side information provided by the relay and the original message from the source to decode the information.
3. Amplify-and-Forward(AF)
   - the relay sends a scaled version of previously received symbol
   - amplification is adjusted according to the relay and the source power constraints

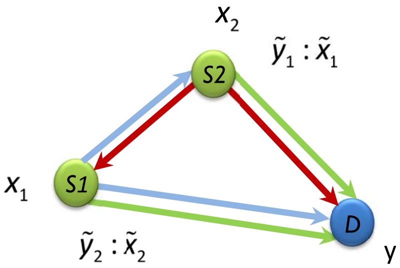
- DF achieves the capacity of degraded relay channel
- DF outperforms CF when relay is close to the source
- CF outperforms DF when the relay is close to destination
- CF always outperforms AF

Coded Cooperation

Relay vs Cooperative Channel



Relay channel



Cooperative channel

- One of the main differences between relaying and user cooperation relates to the different information data injected into the network:
  - Relaying: intermediate node has NO information of its own
  - User cooperation: users relay each other's signals

Coded Cooperation

- Sends different portions of each user's codewords through two (or more) independent fading paths.
- Each user tries to transmit incremental redundancy (IR)/additional parity data for its partner,
- otherwise reverts to non-cooperative mode.
- Cooperation is managed automatically via code design (e.g. ERROR CONTROL CODES)
- No feedback is needed between cooperating users
- Achieves diversity and coding gain

Coded Cooperation

- Half duplex assumption.
- Distinguish two phases:
  1. Broadcast(BC) mode: each user broadcast information to cooperatives users and destination
  2. Multiple access (MAC) mode: cooperative users sends parity data to destination

Coded Cooperation

Key characteristics:

- Cooperation occurs through partition of a user's codeword
  - Level of cooperation quantized in relation to the IR sent by each partner

$$\alpha \triangleq \frac{N_1}{N} = \frac{R}{R_1} \tag{16}$$

  - High degree of flexibility: varying the code rate can adjust to varying channel conditions
- Error detection is employed at the partner to avoid error propagation (eg: Cyclic Redundancy Codes (CRC))
- Similarities with ARQ principle.

Coded Cooperation

- Example of implementation using a Rate Compatible Punctured Convolutional (RCPC) codes
- Puncturing allows to vary cooperation level α

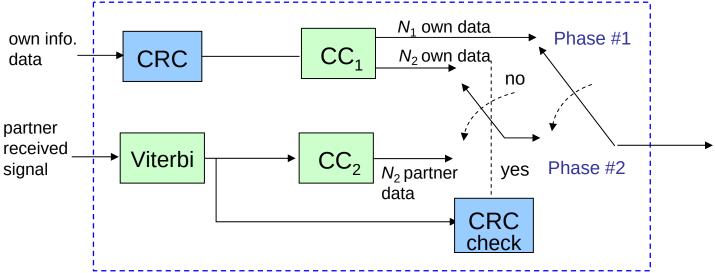
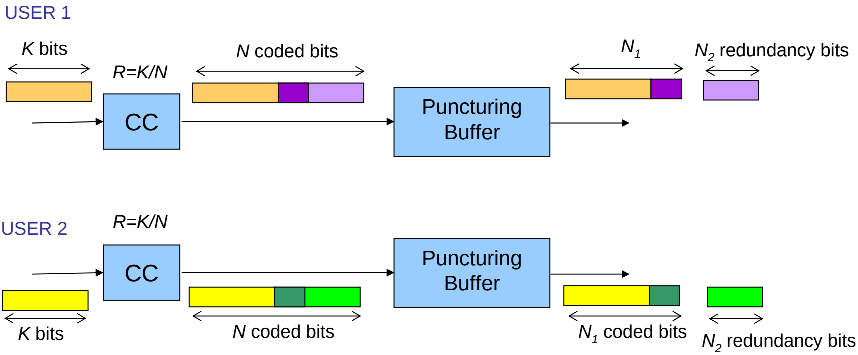


Figure: User implementation block diagram

Coded Cooperation

- Coded information transmitted in each phase. Segmentation of redundancy bits → PUNCTURING



Cases of Coded Cooperation

- In the second phase, users act independently of their own first data block being correctly decoded or not.
- We can distinguish between four scenarios based on decoding results of the first transmission phase:
  1. Both users are able to decode correctly each partner's information
  2. None of the users are able to decode correctly each partner's information
  3. User ♯2 decodes user's ♯1 information, but user ♯1 fails
  4. User ♯1 decodes user's ♯2 information, but user ♯2 fails

- Cooperative overhead for the destination to know which case shall decode: through signalling (additional bits second frame header) or additional complexity at destination.

Outage Analysis for Coded Cooperation

Derivation of **Outage Probability** [HunNos06]

- Next we formulate the outage events for each case.
- First, we establish the baseline for non-cooperative direct transmission in quasi-static fading channel
  - Capacity
$$C(\gamma) = \log_2(1+\gamma)$$
  - Outage Probability
$$P_{OUT} = P_r\{C(\gamma) < R\}$$
  - As a function of the SNR:
$$P_{OUT} = P_r\left\{\gamma < 2^R - 1\right\} = \int_0^{2^R-1} p_\gamma(\gamma)d\gamma$$

$N = N_1 + N_2$	codeword length
$\alpha = N_1/N$	portion of codeword transmitted in phase $\sharp 1$
$1-\alpha$	portion of codeword transmitted in phase $\sharp 2$
R	code rate; $R_1 = R/\alpha$

Outage Analysis for Coded Cooperation

- Need the SNR distribution. For Rayleigh fading channels

$$f(\gamma) = \frac{1}{\bar{\gamma}} \exp\left(-\frac{\gamma}{\bar{\gamma}}\right)$$

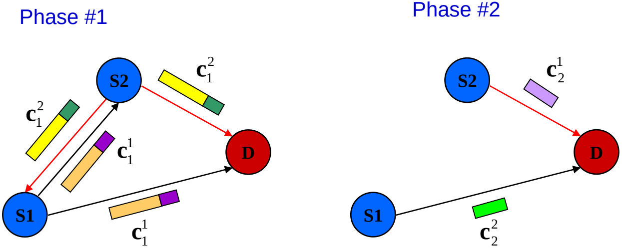
  where $\bar{\gamma}$ is the average SNR

- The outage probability,

$$P_{OUT} = \int_0^{2^R-1} \frac{1}{\bar{\gamma}} \exp\left(-\frac{\gamma}{\bar{\gamma}}\right) d\gamma = 1 - \exp\left(-\frac{2^R-1}{\bar{\gamma}}\right) \tag{17}$$

Outage Analysis for Coded Cooperation
(1) Full cooperation

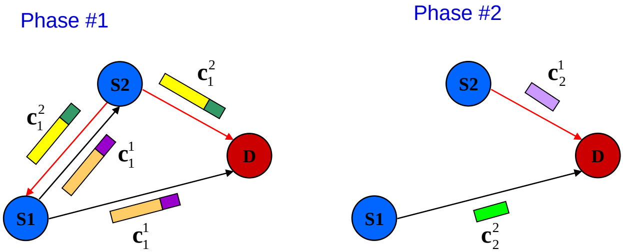
1. Case $\Omega = 1$: Both users able to decode correctly



- Phase ♯1: correct decoding of user's 1 data by user 2 and viceversa

$$C_{12}(\gamma_{12}) > R_1 \Rightarrow \log_2(1+\gamma_{12}) > \frac{R}{\alpha}$$

$$C_{21}(\gamma_{21}) > R_1 \Rightarrow \log_2(1+\gamma_{21}) > \frac{R}{\alpha}$$



- Phase ♯2:the two transmissions can be viewed as parallel conditional Gaussian channels → capacities added

$$C_{1d}\left(\gamma_{1d},\gamma_{2d}|\Omega=1\right)=\alpha\log_2\left(1+\gamma_{1d}\right)+\left(1-\alpha\right)\log_2\left(1+\gamma_{2d}\right)<R$$

$$C_{2d}\left(\gamma_{1d},\gamma_{2d}|\Omega=1\right)=\alpha\log_2\left(1+\gamma_{2d}\right)+\left(1-\alpha\right)\log_2\left(1+\gamma_{1d}\right)<R$$

Outage Probability for Coded Cooperation

Assignment

Sketch the derivation of the outage probability

1. Express outage cases as in Case $\Omega = 1$ for the remaining cases
2. Apply the four cases are disjoint (assumption: SNRs $\gamma_{12}, \gamma_{21}, \gamma_{1d}, \gamma_{2d}$ are mutually independent)
3. Apply SNR distribution to express outage probabilities (integrals) with integration regions

$$\mathcal{A} \equiv \{(\gamma_{1d}, \gamma_{2d}) : (1+\gamma_{1d})^{\alpha} (1+\gamma_{2d})^{1-\alpha} < 2^R\}$$

and

$$\mathcal{B} \equiv \{(\gamma_{1d}, \gamma_{2d}) : (1+\gamma_{1d})^{\alpha} (1+\gamma_{1d}+\gamma_{2d})^{1-\alpha} < 2^R\}$$

4. Hint: for the calculation of integrals

$$\int\int_{\mathcal{A}} \frac{1}{\bar{\gamma}_{1d}} \exp\left(-\frac{\gamma_{1d}}{\bar{\gamma}_{1d}}\right) \frac{1}{\bar{\gamma}_{2d}} \exp\left(-\frac{\gamma_{2d}}{\bar{\gamma}_{2d}}\right) d\gamma_{1d} d\gamma_{2d}$$

Express integration region $\mathcal{A}$ in terms of integration limits for each variable γ_{1d}, γ_{2d},

$$\mathcal{A} \equiv \{(\gamma_{1d}, \gamma_{2d}) : (1+\gamma_{1d})^{\alpha} (1+\gamma_{2d})^{1-\alpha} < 2^R\}$$

$\gamma_{2d} < \frac{2^{R/(1-\alpha)}}{(1+\gamma_{1d})^{\alpha/(1-\alpha)}} - 1 \equiv a$

$\gamma_{2d} > 0$

$$\frac{2^{R/(1-\alpha)}}{(1+\gamma_{1d})^{\alpha/(1-\alpha)}} > 1$$

- Then

$$\int_0^{2^{R/\alpha}-1} \frac{1}{\bar{\gamma}_{1d}} \exp\left(-\frac{\gamma_{1d}}{\bar{\gamma}_{1d}}\right) \left(\int_0^a \frac{1}{\bar{\gamma}_{2d}} \exp\left(-\frac{\gamma_{2d}}{\bar{\gamma}_{2d}}\right) d\gamma_{2d}\right) d\gamma_{1d}$$

Outage probability for user 1

$$P_{OUT}^{(1)} = \exp\left(\frac{1-2^{R/\alpha}}{\bar{\gamma}_{21}}\right)\left[1-\exp\left(\frac{1-2^{R/\alpha}}{\bar{\gamma}_{1d}}\right)-\exp\left(\frac{1-2^{R/\alpha}}{\bar{\gamma}_{12}}\right)\Psi_1(\bar{\gamma}_{1d},\bar{\gamma}_{2d},R,\alpha)\right]$$
$$+\quad\left(1-\exp\left(\frac{1-2^{R/\alpha}}{\bar{\gamma}_{21}}\right)\right)\left[1-\exp\left(\frac{1-2^{R}}{\bar{\gamma}_{1d}}\right)-\exp\left(\frac{1-2^{R/\alpha}}{\bar{\gamma}_{12}}\right)\Psi_2(\bar{\gamma}_{1d},\bar{\gamma}_{2d},R,\alpha)\right]$$

where

$$\Psi_1(\bar{\gamma}_{1d},\bar{\gamma}_{2d},R,\alpha) = \int_0^{2^R/\alpha-1}\frac{1}{\bar{\gamma}_{1d}}\exp\left(-\frac{\gamma_{1d}}{\bar{\gamma}_{1d}}-\frac{a}{\bar{\gamma}_{2d}}\right)d\gamma_{1d}$$

and

$$a = \frac{2^{R/(1-\alpha)}}{(1+\gamma_{1d})^{\alpha/(1-\alpha)}}-1$$

$$\Psi_2(\bar{\gamma}_{1d},\bar{\gamma}_{2d},R,\alpha) = \int_0^{2^R-1}\frac{1}{\bar{\gamma}_{1d}}\exp\left(-\frac{\gamma_{1d}}{\bar{\gamma}_{1d}}-\frac{b}{\bar{\gamma}_{2d}}\right)d\gamma_{1d}$$

and

$$b = \frac{2^{R/(1-\alpha)}}{(1+\gamma_{1d})^{\alpha/(1-\alpha)}}-1-\gamma_{1d}$$

Outage Probability for Coded Cooperation

For the particular case of reciprocal inter-user channels $\gamma_{12} = \gamma_{21}$,

$$\begin{aligned} P_{OUT}^{(1)} &= \exp\left(\frac{1-2^{R/\alpha}}{\bar{\gamma}_{12}}\right)\left[1-\exp\left(\frac{1-2^{R/\alpha}}{\bar{\gamma}_{1d}}\right)-\Psi_1(\bar{\gamma}_{1d},\bar{\gamma}_{2d},R,\alpha)\right]+ \\ &+ \left[1-\exp\left(\frac{1-2^{R/\alpha}}{\bar{\gamma}_{12}}\right)\right]\left[1-\exp\left(\frac{1-2^{R/\alpha}}{\bar{\gamma}_{1d}}\right)\right] \end{aligned}$$

- Outage probability for coded cooperation depends on: mean SNR, code rate and cooperation level α
- Optimization of design parameter α is complex. May be obtained through iteration.

Asymptotic analysis - Diversity gain

Asymptotic analysis in the high SNR regime

- The approach: re-parameterize the mean SNR (decouple user transmit power from channel impairments)
  - $\bar{\gamma}_T$, user transmit power over receive noise power
  - λ_{ij}, accounting for large scale effects (path loss and shadowing)

$$\bar{\gamma}_{ij} = \bar{\gamma}_T \lambda_{ij} \text{ for } i, j = 1, 2, d$$

- $\bar{\gamma}_T \to \infty$, diversity order $\to$ smallest exponent for $\frac{1}{\bar{\gamma}_T}$
- Sketch of derivation: Taylor series expansion of exponential terms

Asymptotic analysis - Diversity gain

Asymptotic analysis in the high SNR regime

1. Independent inter-user channel

$$P_{OUT} = \frac{1}{\bar{\gamma}_T^2}\left(\frac{\left(2^{R/\alpha}-1\right)^2}{\bar{\gamma}_{1d}\bar{\gamma}_{12}} + \frac{f(R,\alpha)}{\bar{\gamma}_{1d}\bar{\gamma}_{2d}}\right) + \mathcal{O}\left(\frac{1}{\bar{\gamma}_T^3}\right)$$

Asymptotic analysis - Diversity gain

Asymptotic analysis in the high SNR regime

1. Independent inter-user channel

$$P_{OUT} = \frac{1}{\bar{\gamma}_T^2}\left(\frac{\left(2^{R/\alpha}-1\right)^2}{\bar{\gamma}_{1d}\bar{\gamma}_{12}} + \frac{f(R,\alpha)}{\bar{\gamma}_{1d}\bar{\gamma}_{2d}}\right) + \mathcal{O}\left(\frac{1}{\bar{\gamma}_T^3}\right)$$

2. Reciprocal inter-user channel, $\gamma_{12} = \gamma_{21}$

$$P_{OUT} = \frac{1}{\bar{\gamma}_T^2}\left(\frac{\left(2^{R}-1\right)\left(2^{R/\alpha}-1\right)}{\bar{\gamma}_{1d}\bar{\gamma}_{12}} + \frac{f(R,\alpha)}{\bar{\gamma}_{1d}\bar{\gamma}_{2d}}\right) + \mathcal{O}\left(\frac{1}{\bar{\gamma}_T^3}\right)$$

$$f(R,\alpha) = \begin{cases} \left(\frac{\alpha}{1-2\alpha}\right)2^{R/\alpha} - \left(\frac{1-\alpha}{1-2\alpha}\right)2^{R/(1-\alpha)} + 1 & \alpha \neq \frac{1}{2} \\ R \cdot 2^{2R+1} \cdot \ln 2 - 2^{2R} + 1 & \alpha = \frac{1}{2} \end{cases}$$

Asymptotic analysis - Diversity gain

Asymptotic analysis in the high SNR regime

1. Independent inter-user channel

$$P_{OUT} = \frac{1}{\bar{\gamma}_T^2}\left(\frac{\left(2^{R/\alpha}-1\right)^2}{\bar{\gamma}_{1d}\bar{\gamma}_{12}} + \frac{f(R,\alpha)}{\bar{\gamma}_{1d}\bar{\gamma}_{2d}}\right) + \mathcal{O}\left(\frac{1}{\bar{\gamma}_T^3}\right)$$

2. Reciprocal inter-user channel, $\gamma_{12} = \gamma_{21}$

$$P_{OUT} = \frac{1}{\bar{\gamma}_T^2}\left(\frac{\left(2^{R}-1\right)\left(2^{R/\alpha}-1\right)}{\bar{\gamma}_{1d}\bar{\gamma}_{12}} + \frac{f(R,\alpha)}{\bar{\gamma}_{1d}\bar{\gamma}_{2d}}\right) + \mathcal{O}\left(\frac{1}{\bar{\gamma}_T^3}\right)$$

$$f(R,\alpha) = \begin{cases} \left(\frac{\alpha}{1-2\alpha}\right)2^{R/\alpha} - \left(\frac{1-\alpha}{1-2\alpha}\right)2^{R/(1-\alpha)} + 1 & \alpha \neq \frac{1}{2} \\ R \cdot 2^{2R+1} \cdot \ln 2 - 2^{2R} + 1 & \alpha = \frac{1}{2} \end{cases}$$

Coded cooperation achieves **full diversity**(=2 in the example)

Asymptotic analysis - Diversity gain

Extension to multiple partners ($n > 2$), with protocol

1. user i transmits, all the other users listen
2. only those users who correctly decode user i's signal send extra information about user i's frame to destination

Has also been analyzed and full diversity achievement demonstrated [HunSan06]

$$\bar{\gamma}_T \rightarrow \infty : P_{OUT} \propto \mathcal{O}(\frac{1}{\bar{\gamma}_T^n})$$

Performance of Coded Cooperation

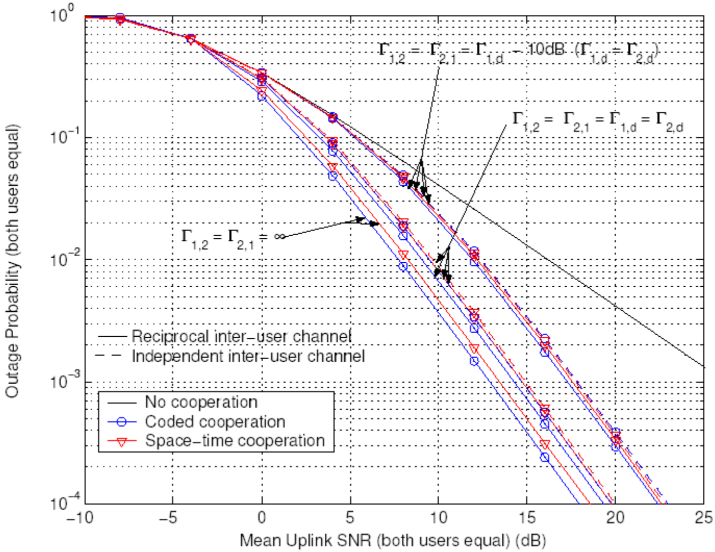
Example

- Rate R=1/2
- Assumes inter-user links have the same average channel quality ($\bar{\gamma}_{12} = \bar{\gamma}_{21}$) and considers the case where the uplink mean SNR is equal for both users ($\bar{\gamma}_{1d} = \bar{\gamma}_{2d}$) $\Rightarrow P_{OUT}^{(1)} = P_{OUT}^{(2)}$

Performance features:

- Cooperation improves performance even for poor inter-user link quality
- When direct link and inter-user link exhibit the same channel quality cooperation brings up most of the achievable gains
- When inter-user channel quality increases over direct link, offers small additional improvement (in the limit, $\bar{\gamma}_{12} \to \infty$, $\approx$ 2dB)
- Reciprocal inter-user channel $\to$ slightly better performance than independent channels.

Performance of Coded Cooperation



Source ©[HunSan06]

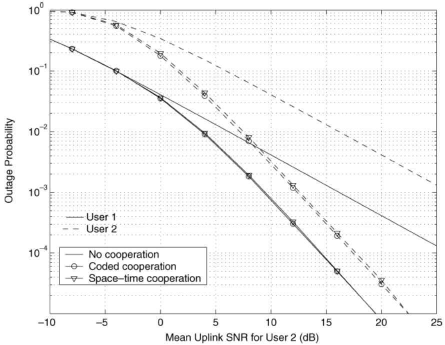
Performance of Coded Cooperation

Example

- This case considers the scenario where uplink channel quality between cooperative users is different, e.g $\bar{\gamma}_{1d} = \bar{\gamma}_{2d} + 10$dB (R=1/2, $\bar{\gamma}_{12} = \bar{\gamma}_{21}$)

Performance results:

- As expected user 2 improves its performance over non-cooperative transmission
- but so does user 1 besides the poorer quality of its partner's link.



Source ©[HunSan06]

Distributed Coding Schemes

Distributed Coding Schemes

Code examples

1. Rate Compatible Punctured Convolutional (RCPC) code
2. Distributed Turbo Codes (DTC)
3. Distributed Turbo Codes with Soft Information Relaying (DTC-SIR)
4. Generalized Distributed Turbo Codes (GDTC)
5. Distributed Low Density Parity Check Codes (DLDPC)

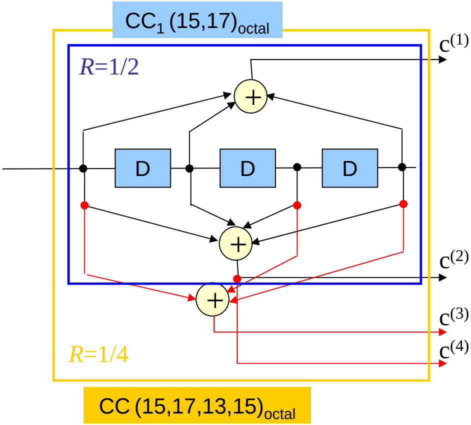
6. Network Coding (multiuser multihop networks)

Convolutional Codes

- The RCPC codes used by [HunNos04] where not optimized for cooperation.
- Stefanov and Erkip [SteErk04] propose the design of channel codes suitable for cooperative transmissions based on code design for block fading channels. From the point of view of the destination the channel follows a block fading model rather than slow/quasi-static fading channel. However, the cooperative scenario introduces additional constraints:
  - For cooperation to occur often: effective code R_1 should be a good code in quasi-static fading ($N = N_1 + N_2$)
  - But also the overall code R should be a **good code** for quasi-static fading (partner unable to decode correctly $\Rightarrow$ non cooperative mode) and block fading (partner able to decode correctly)

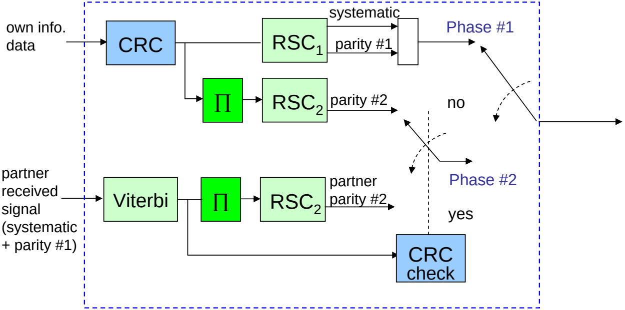
An example

- $R = \frac{1}{2}$ $CC_1(15,17)_{octal}$ best code for inter-user channel
- $R = \frac{1}{4}$ $CC(15,17,13,17)_{octal}$ good performance for quasi-static direct transmission (no cooperation); diversity gain=2 and good coding gain in block fading channels.



Distributed Turbo Codes

- Encoding process takes place at each cooperative node



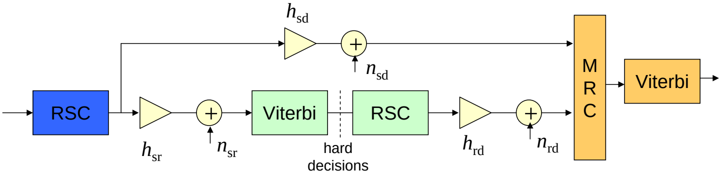
- Cooperative nodes and destination have the same interleaver
- Each user transmit its partner's parity bits in the second frame using all the available power
- Rate compatible punctured codes introduce flexibility in the level of cooperation ⇒ turbo decoding required at the cooperative node
- DTC also considered for the relay channel

Distributed Turbo Codes

- **Decode and Forward** approach:
  - Code: Repetition coding
  - Exploits receive diversity: maximum ratio combining at the receiver
  - Diversity gain
- **Distributed TC** approach:
  - Code: TC based on recursive systematic convolutional codes (RSC)
  - Turbo principle at the receiver
  - Diversity and coding gain (In general diversity gain proportional to number of cooperative nodes)

Distributed Turbo Codes

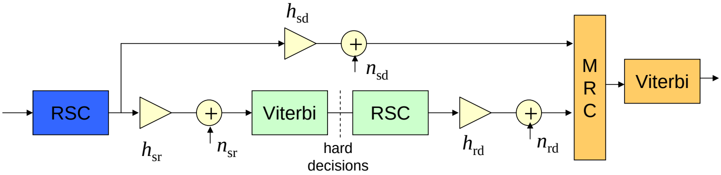
- Block diagram **Decode and Forward** approach:



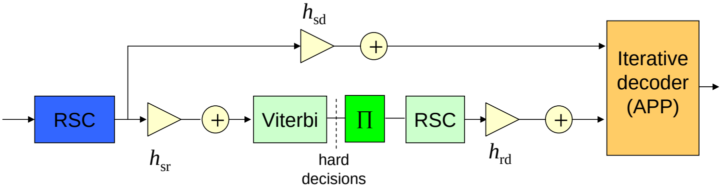
Analysis assume quasi-static Rayleigh fading channel . Each link has a constant fading level during N symbols.

Distributed Turbo Codes

- Block diagram **Decode and Forward** approach:



- Block diagram **Distributed TC** approach:



Analysis assume quasi-static Rayleigh fading channel . Each link has a constant fading level during N symbols.

Iterative decoding of DTC

- Decoder at destination $\rightarrow$ standard turbo decoder
- Remark regarding complexity: constituent codes can be very simple (few states RSC code). This is a property of Turbo Codes in general: Turbo Codes perform better with relatively low complexity constituent codes

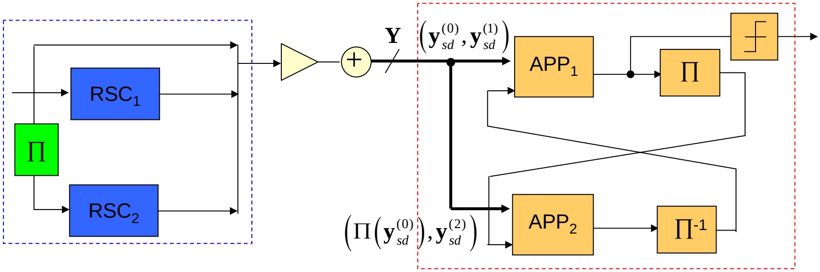


Figure: Encoding/Decoding block diagram for conventional PCCC.

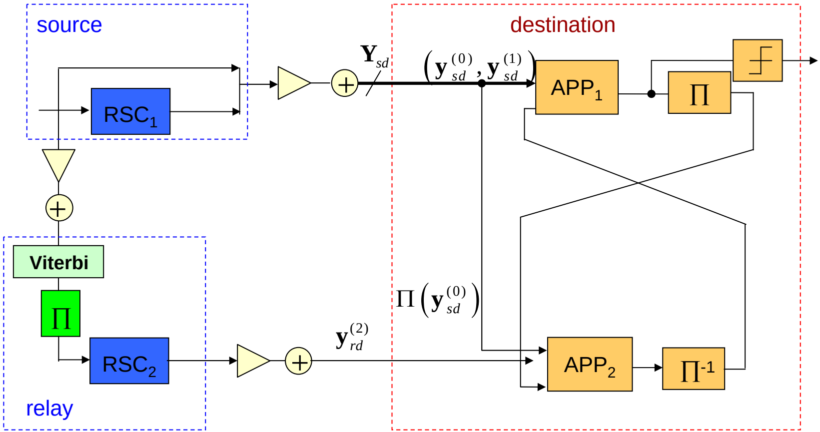
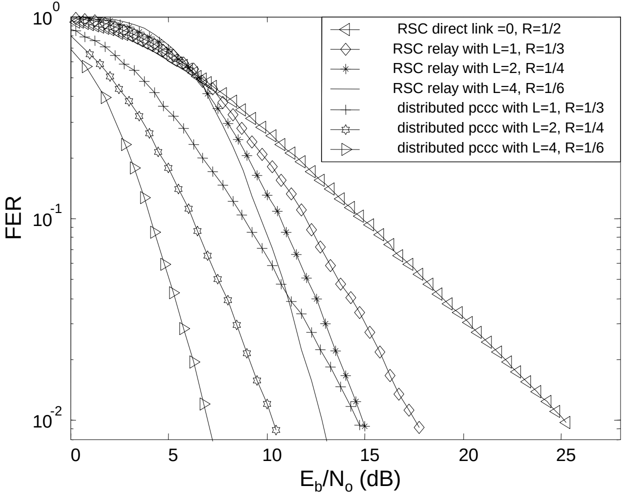


Figure: Encoding/Decoding block diagram distributed turbo code.

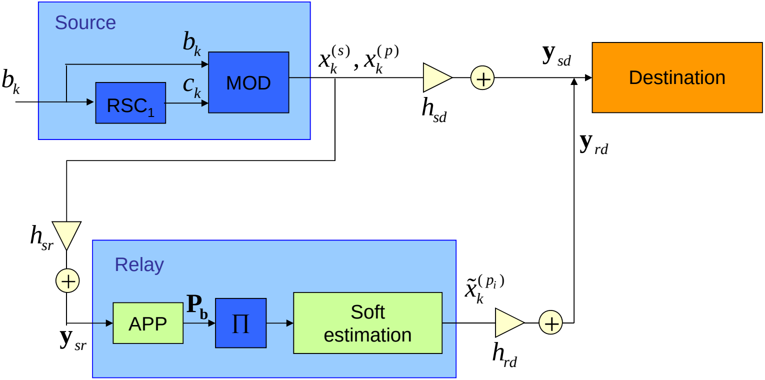
Performance of DTC for the relay channel

An example (perfect source-to-relay channels assumed)



Source ©[ValZha03]

Distributed Turbo Codes with Soft Information



- Soft decisions at relay instead of hard decisions
- Protocol known under Soft Information Relaying (**SIR**)
- Scheme works with *a posteriori* probabilities (APP)
- Achieves full diversity order ($=N$, number of cooperating nodes), plus improves coding gain

Distributed Turbo Codes with Soft Information

Processing performed at the relay includes 3 main steps:

1. Calculation of APPs of the systematic data $P_r\left\{b_k = a|\mathbf{y}\right\}, a \in \{0, 1\}$

Distributed Turbo Codes with Soft Information

Processing performed at the relay includes 3 main steps:

1. Calculation of APPs of the systematic data $P_r\left\{b_k = a|\mathbf{y}\right\}, a \in \{0,1\}$
2. Calculation of APPs associated with the interleaved data $P_r\left\{b_k^{'} = a|\mathbf{y}\right\}, a \in \{0,1\}$ and $P_r\left\{c_k^{'} = a|\mathbf{y}\right\}, a \in \{0,1\}$

Distributed Turbo Codes with Soft Information

Processing performed at the relay includes 3 main steps:

1. Calculation of APPs of the systematic data $P_r\left\{b_k = a|\mathbf{y}\right\}, a \in \{0,1\}$
2. Calculation of APPs associated with the interleaved data $P_r\left\{b_k^{'} = a|\mathbf{y}\right\}, a \in \{0,1\}$ and $P_r\left\{c_k^{'} = a|\mathbf{y}\right\}, a \in \{0,1\}$
3. Calculation of parity symbols soft estimates of the interleaved data $\tilde{x}_k^{'}$

Distributed Turbo Codes with Soft Information

1. **Calculation of the APPs for the systematic data** as in conventional turbo decoding (BCJR algorithm)

$$\mathbf{P_b} : P_r \left\{ b_k = a | \mathbf{y}_{sr} \right\}, k = 1, \ldots, K \qquad a = 0, 1$$

$\mathbf{y}_{sr}$ is the received signal sequence at the relay

$$P_r \left\{ b_k = a | \mathbf{y}_{sr} \right\} = \eta \sum_{m,m'=0; b_k=a}^{m,m'=M_S-1} \alpha_{k-1}(m')\beta_k(m)\gamma_k(m, m')$$

η normalization factor such that

$$\sum_a P_r \left\{ b_k = a | \mathbf{y}_{sr} \right\} = 1$$

$\alpha_k(m')$, $\beta_k(m)$, feedforward and feedback recursive variable

$\gamma_k(m, m')$ branch metric

$$\gamma_k(m, m') = \exp\left(-\frac{\left\| \mathbf{y}_{sr}(k) - \sqrt{P_{sr}} h_{sr} \mathbf{x}(k) \right\|^2}{N_0} \right)$$

2. Calculation of the APPs for the interleaved data

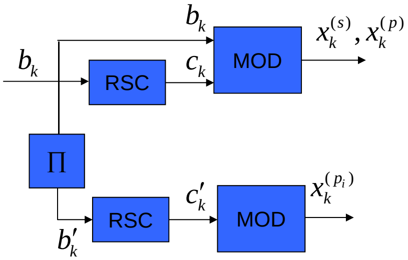
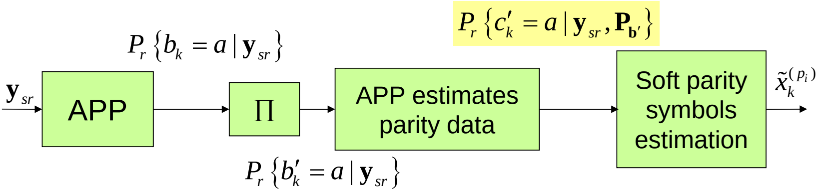


Figure: PCCC encoder

- Vector $\mathbf{b}'$ denotes interleaved information bits.
- Vector $\mathbf{c}'$ denotes coded bits (parity bits) associated to $\mathbf{b}'$.
- Assumed infinite length interleavers, $\mathbf{P}_{\mathbf{b}'} = \prod(\mathbf{P}_{\mathbf{b}})$

$$\mathbf{P}_{\mathbf{b}'} : P_r \left\{ b_k' = a | \mathbf{y}_{sr} \right\}, k = 1, \ldots, K \qquad a = 0, 1$$

2. **Calculation of the APPs for the interleaved data** cont.



Distributed Turbo Codes with Soft Information

2. **Calculation of the APPs for the interleaved data** cont.

- Develop a recursive algorithm similar to BCJR that computes APP of $\mathbf{c}'$ parity bits for the interleaved data

$$
\begin{aligned}
P_r\left\{c_k^{'}=a|\mathbf{y}_{sr},\mathbf{P}_{\mathbf{b}'}\right\} &= \sum_{m\in\mathcal{U}(c_k^{'}=a)} P_r\left\{b_k^{'}=w|\mathbf{y}_{sr},\mathbf{P}_{\mathbf{b}'},S_{k-1}=m\right\}\cdot P_r\left\{S_{k-1}=m|\mathbf{y}_{sr},\mathbf{P}_{\mathbf{b}'}\right\} \\
&= \sum_{m\in\mathcal{U}(c_k^{'}=a)} P_r\left\{b_k^{'}=w|\mathbf{y}_{sr}\right\}\cdot P_r\left\{S_{k-1}=m|\mathbf{y}_{sr},\mathbf{P}_{\mathbf{b}'}\right\}
\end{aligned}
$$

$$
\begin{aligned}
P_r\left\{S_k=m|\mathbf{y}_{sr},\mathbf{P}_{\mathbf{b}'}\right\} &= \sum_{m'} P_r\left\{S_k=m|S_{k-1}=m',\mathbf{y}_{sr},\mathbf{P}_{\mathbf{b}'}\right\}\cdot P_r\left\{S_{k-1}=m'|\mathbf{y}_{sr},\mathbf{P}_{\mathbf{b}'}\right\} \\
&= \sum_{m'} P_r\left\{b(m,m')|\mathbf{y}_{sr}\right\}\cdot P_r\left\{S_{k-1}=m'|\mathbf{y}_{sr},\mathbf{P}_{\mathbf{b}'}\right\}
\end{aligned}
$$

$\mathcal{U}(c_k^{'}=a)$ set of branches for which the output parity symbol is equal to a

Distributed Turbo Codes with Soft Information

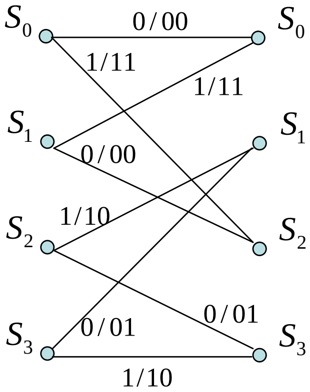
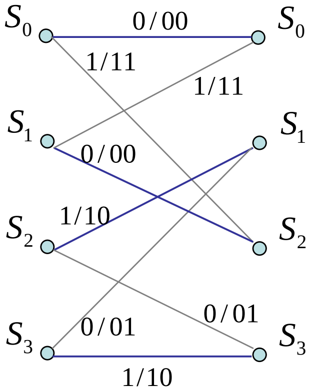


Figure: Example 4-state trellis

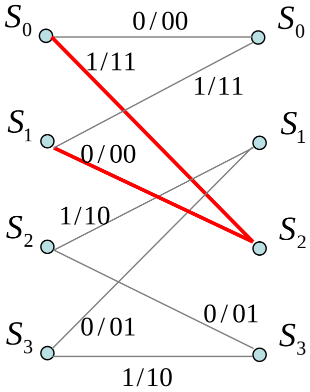
Distributed Turbo Codes with Soft Information



$$P_r\left\{c_k^{'}=a|\mathbf{y}_{sr},\mathbf{P}_{\mathbf{b}'}\right\}=\sum_{m\in\mathcal{U}(c_k^{'}=a)}P_r\left\{b_k^{'}=w|\mathbf{y}_{sr}\right\}\cdot P_r\left\{S_{k-1}=m|\mathbf{y}_{sr},\mathbf{P}_{\mathbf{b}'}\right\}$$

$$\begin{aligned}P_r\left\{c_k^{'}=0|\mathbf{y}_{sr},\mathbf{P}_{\mathbf{b}'}\right\} &= P_r\left\{b_k^{'}=0|\mathbf{y}_{sr}\right\}P_r\left\{S_{k-1}=S_0|\mathbf{y}_{sr},\mathbf{P}_{\mathbf{b}'}\right\}+P_r\left\{b_k^{'}=0|\mathbf{y}_{sr}\right\}P_r\left\{S_{k-1}=S_1|\mathbf{y}_{sr},\mathbf{P}_{\mathbf{b}'}\right\}\\ &+ P_r\left\{b_k^{'}=1|\mathbf{y}_{sr}\right\}P_r\left\{S_{k-1}=S_2|\mathbf{y}_{sr},\mathbf{P}_{\mathbf{b}'}\right\}+P_r\left\{b_k^{'}=1|\mathbf{y}_{sr}\right\}P_r\left\{S_{k-1}=S_3|\mathbf{y}_{sr},\mathbf{P}_{\mathbf{b}'}\right\}\end{aligned}$$

Distributed Turbo Codes with Soft Information



$$P_r\left\{S_k = m|\mathbf{y}_{sr}, \mathbf{P}_{\mathbf{b}'}\right\} = \sum_{m'} P_r\left\{b(m,m')|\mathbf{y}_{sr}\right\} \cdot P_r\left\{S_{k-1} = m'|\mathbf{y}_{sr}, \mathbf{P}_{\mathbf{b}'}\right\}$$

$$P_r\left\{S_k = S_2|\mathbf{y}_{sr}, \mathbf{P}_{\mathbf{b}'}\right\} = P_r\left\{b_k' = 1|\mathbf{y}_{sr}\right\} P_r\left\{S_{k-1} = S_0|\mathbf{y}_{sr}, \mathbf{P}_{\mathbf{b}'}\right\} + P_r\left\{b_k' = 0|\mathbf{y}_{sr}\right\} P_r\left\{S_{k-1} = S_1|\mathbf{y}_{sr}, \mathbf{P}_{\mathbf{b}'}\right\}$$

3. **Calculation of soft estimates**
   - Linear combination of APPs on parity bits
   - e.g BPSK $0 \rightarrow 1$, $1 \rightarrow -1$ the soft estimate is given by

$$\tilde{x}_k^{(p_i)} = 1 \cdot P_r \left\{ c_k^{'} = 0 | \mathbf{P}_{\mathbf{b}'} \right\} - 1 \cdot P_r \left\{ c_k^{'} = 1 | \mathbf{P}_{\mathbf{b}'} \right\}$$

Performance of DTC with SIR

Example parameters: (Source [LiVuc05])

- Quasi-static Rayleigh fading channel
- BPSK modulation
- Frame size 130 symbols
- Code rate $R = \frac{1}{2}$
- Generator polynomials of component convolutional code $(1, \frac{5}{7})_{octal}$

Performance example of DTC with SIR - BER

- DTC-SIR Distributed Turbo Code with Soft Information Relaying
- DTC Distributed Turbo Code
- DTC-SIR Distributed Turbo Code with ARQ between source-relay (maximum number of retransmissions = 3)

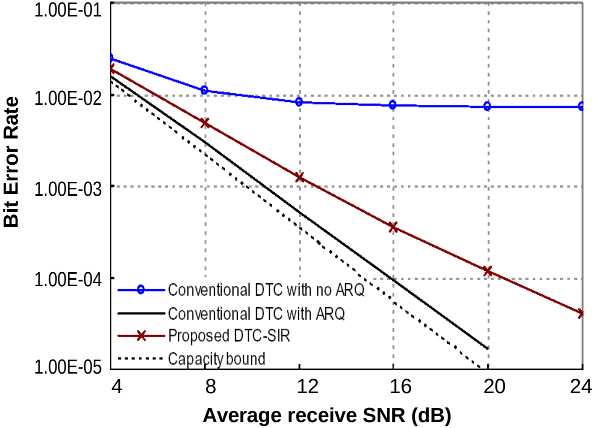


Figure: BER at source to relay channel reliability $\gamma_{sr} = 10\text{dB}$ [LiVuc06]

Performance example of DTC with SIR - BER

- DTC-SIR Distributed Turbo Code with Soft Information Relaying
- DTC Distributed Turbo Code
- DTC-SIR Distributed Turbo Code with ARQ between source-relay (maximum number of retransmissions = 3)

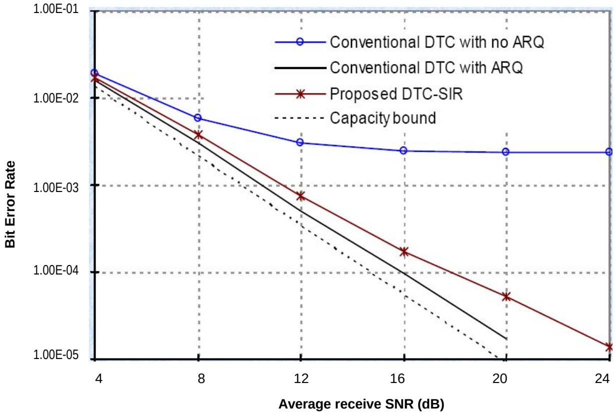


Figure: BER at source to relay channel reliability $\gamma_{sr} = 15$dB [LiVuc06]

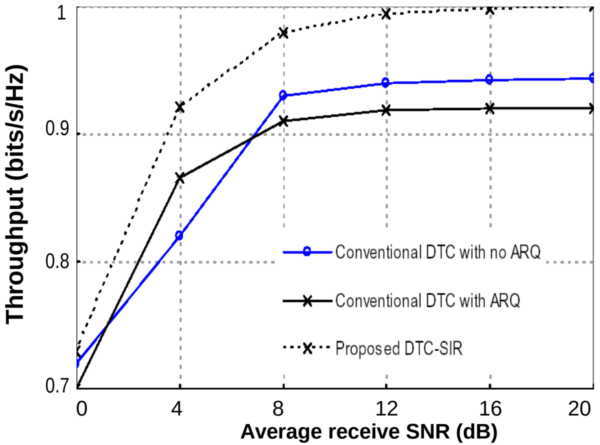


Figure: Throughput at source to relay channel reliability $\gamma_{sr} = 10$dB [LiVuc06]

Distributed Low Density Parity Check Codes

Distributed LDPC

Description of LDPC codes

- Sparse graph codes
- Parity check matrix of a rate $1/2$ code
- $\mathbf{H} < M \times N >$, N **variable nodes**, M **check nodes**

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

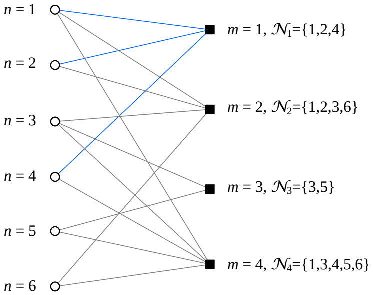


Figure: Bipartite graph

Distributed LDPC

Description of LDPC codes

- Sparse graph codes
- Parity check matrix of a rate 1/2 code
- **H** $< M \times N >$, N **variable nodes**, M **check nodes**

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

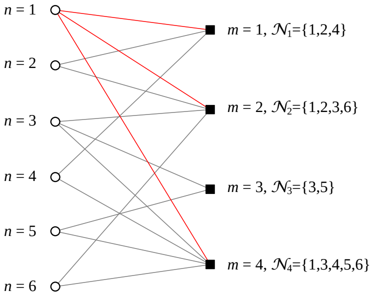


Figure: Bipartite graph

- Maximum-likelihood decoding $\hat{\mathbf{c}} = \max_{\mathbf{c}\in\mathcal{C}} \Pr[\mathbf{c}|\mathbf{y}, \mathbf{Hc}^T = 0]$
- Sum-product algorithm approaches ML for graph with no cycles

Some definitions:

- $\mathcal{N}_m \triangleq \{n : H_{mn} = 1\}$
- $\mathcal{N}_{m\backslash n} \triangleq \{n \neq m : H_{mn} = 1\}$
- $\mathcal{M}_n \triangleq \{m : H_{mn} = 1\}$
- $\mathcal{M}_{n\backslash m} \triangleq \{m \neq n : H_{mn} = 1\}$
- Row weight $w_r(m) = |\mathcal{N}_m|$, column weight $w_r(m) = |\mathcal{M}_n|$
- Parity check syndrome $s_m = \sum_{n=1}^{N} H_{mn}c_n = \sum_{n\in\mathcal{N}_m} c_n$

- We define the following probabilities:

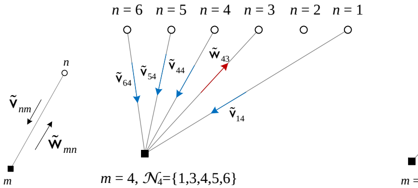
$$\begin{aligned} v_n(b) &= \Pr[c_n = b|\mathbf{y}, \mathcal{S}_n] \text{ pseudoposterior probability} \\ v_{nm}(b) &= \Pr[c_n = b|\mathbf{y}, \mathcal{S}_{nm}] \text{ message variable} \rightarrow \text{check} \\ w_{mn}(b) &= \Pr[s_m = 0|c_n = b, \mathbf{y}] \text{ message check} \rightarrow \text{variable} \\ p_n(b) &\triangleq \Pr[c_n = 0|\mathbf{y}] = \Pr[c_n = 0|y_n] \text{ channel input to each variable node} \\ &\text{with } b \in \mathbb{F}_2 \\ &\text{the event } \mathcal{S}_n\{s_m = 0, \forall m \in \mathcal{M}_n\} \\ &\text{and the event } \mathcal{S}_{nm}\{s'_m = 0, \forall m' \in \mathcal{M}_{n\backslash m}\} \end{aligned}$$

- The algorithm can be simplified by defining the messages to be passed as L-values:

$$
\begin{aligned}
L_n &\triangleq \ln \frac{p_n(0)}{p_n(1)} \\
\tilde{v}_{nm} &\triangleq \ln \frac{v_{nm}(0)}{v_{nm}(1)} \quad \text{variable} \rightarrow \text{check node} \\
\tilde{w}_{mn} &\triangleq \ln \frac{w_{mn}(0)}{w_{mn}(1)} \quad \text{check} \rightarrow \text{variable node}
\end{aligned}
$$

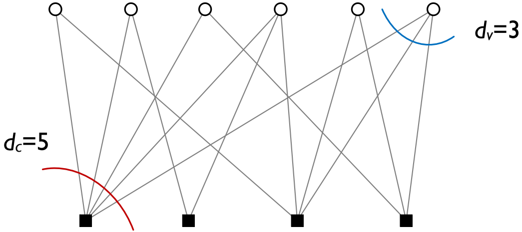
- The basic sum-product algorithm with L-values

Initialize
for $n = 1, 2 \ldots, N$ **do**
    for $m \in \mathcal{M}_n$ **do**
        $\tilde{v}_{nm} = L_n$
    end for
end for
for $i = 1, 2, \ldots, i_{\max}$ **do**
    check node update
    for $m = 1, 2, \ldots, M$ **do**
        for $n \in \mathcal{N}_m$ **do**
            $\tilde{w}_{mn} = 2 \cdot \operatorname{atanh}\left(\prod_{n' \in \mathcal{N}_{m \setminus n}} \tanh \frac{\tilde{v}_{n'm}}{2}\right)$
        end for
    end for
    variable node update
    for $n = 1, 2, \ldots, N$ **do**
        for $m \in \mathcal{M}_n$ **do**
            $\tilde{v}_{nm} = L_n + \sum_{m' \in \mathcal{M}_{n \setminus m}} \tilde{w}_{m'n}$
        end for
        $\tilde{v}_n = L_n + \sum_{m \in \mathcal{M}_n} \tilde{w}_{mn}$
        $\hat{c}_n = 1[\tilde{v}_n < 0]$
    end for
    if $\mathbf{H}\hat{\mathbf{c}}^{\mathrm{T}} = \mathbf{0}$ **then**
        break
    end if
end for



Design rules for LDPC codes

- Density evolution
- d_v maximum number of edges connected to a variable node
- d_c maximum number of edges connected to a check node



Distributed LDPC

LDPC code design rules

- Code described in terms of Degree profiles:

$$\lambda(x) = \sum_{i=2}^{d_v} \lambda_i x^{i-1} \text{ and } \rho(x) = \sum_{i=2}^{d_c} \rho_i x^{i-1}$$

- Almost all codes with the same profile have similar decoding performance (in the limit of blocklength and infinite iterations)

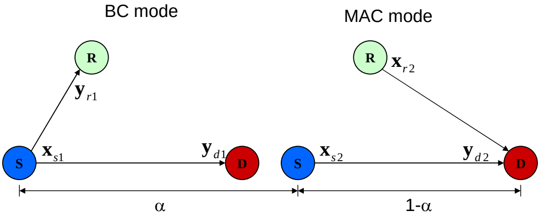
Distributed LDPC

LDPC code design rules

- Decoding of LDPC codes by message passing algorithms ⇒ **belief propagation**(sum-product algorithm)
- **Density evolution** predicts the outcome of the message passing decoding by tracking message probability densities over successive iterations ⇒ discovers noise threshold (below threshold successful decoding with high probability for a randomly chosen code with that profile)
- Use of density evolution to search for good codes

Distributed LDPC

- Chakrabarti et al. [ChaBay07] introduce **Distributed LDPC** codes for the relay channel
- Assume two-phase transmission:



- What are the essential steps in building an optimum distributed coding scheme?
- Approach: Modify code design rules for LDPC, taking into account design constraints derived from the cooperative scenario

Distributed LDPC

Code design challenge for relay channel:

- Joint optimization of multiple constituent LDPC code profiles
- Observed that codebooks can be completely correlated $r = 0$ or independent $r = 1$ without significant rate loss
- Definition of *correlation* r: relays sends $\mathbf{c}_{rd}$ and the source $r\mathbf{c}_{rd} + (1-r)\mathbf{c}_{sd}$ where $\mathbf{c}_{rd}, \mathbf{c}_{rd}$ are (binary) codewords from independent codebooks $\mathcal{C}_{rd}$ and $\mathcal{C}_{sd}$, respectively
- Relay code profile optimization, requires building two LDPC codes that are both good single-user codes of rates R_{sr} and R_{sd}, such that the bipartite graph of $\mathcal{C}_{sr}$ is a subgraph of $\mathcal{C}_{sd}$
- This translates into additional constraints on degree distributions for the density evolution algorithm

What's next

- We have seen that through proper code design distributed coding can achieve both **spatial diversity and coding gain**
- Most of the distributed coding schemes have been developed based on conventional channel coding schemes (Turbo Codes, LDPC).
- Detection errors at the relay/cooperative user do have impact on performance
- There is no accurate analytical representation to model decoding errors
- Optimum code design for cooperative channel still an open issue
- Scalability to multicast multi-hop networks ⇒ Network Coding

Coding at upper layers

- Motivation and definition
- Rateless codes
- Random coding and analysis

The basic idea

- In an end-to-end network connection, some packets get lost for a variety of reasons:
  - Errors from lower layers (typical in wireless, less in wired networks)
  - Buffer overflows
  - Congestion control mechanisms:"Random early detection" (RED) selectively drops some packets to reduce TCP congestion window ...
- We model lost packets as channel erasures

Introduction to Network Coding

- Channel erasures trigger packet retransmissions
- We wish to find a mechanism that avoids the need for retransmissions
    - Reduce delay
    - Increase robustness
    - Reduce signaling overhead (acks/nacks)
- We do so by including redundancy in the transmitted data, in a distributed way, over the network

The Broadcast/Multicast Problem

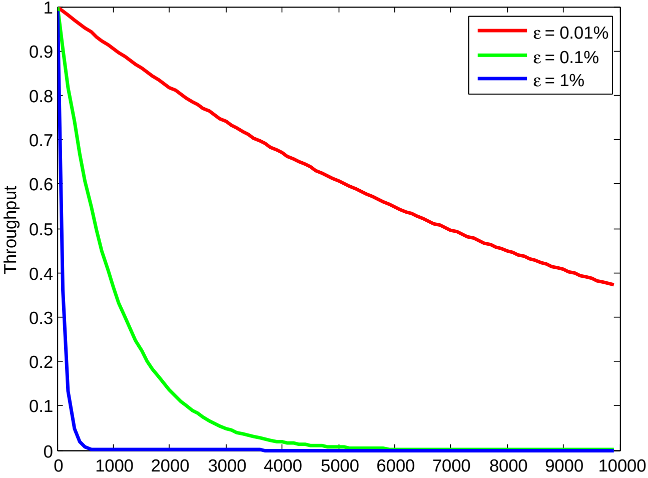
- Server needs to transmit software update to multiple clients
- Each terminal connected through independent erasure channel with erasure probability ϵ
- Throughput, 1 terminal:

$$\eta = P(\bar{E}) = 1 - P(E) = 1 - \epsilon$$

- Throughput, N terminals :

$$\eta = P(\bar{E}_1) \bigcap P(\bar{E}_2) \bigcap \ldots \bigcap P(\bar{E}_N) = (1-\epsilon)^N$$

The Broadcast/Multicast Problem



- For a large number of clients, all packets need to be retransmitted

Definition

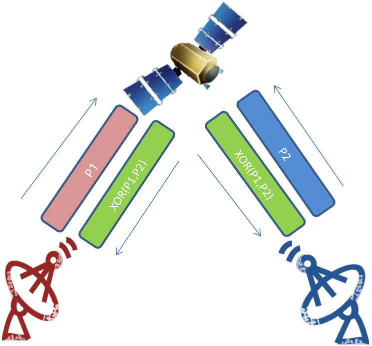
- A basic definition of network coding would be:

  Encoding scheme for transmission over multiple network edges, where intermediate nodes may perform some operations on the message content

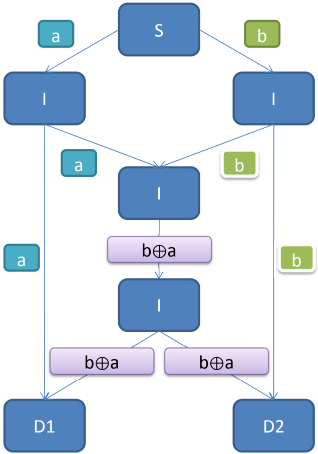
- Network coding is performed at link, network, or upper layers,
  - It is performed over a binary field $\mathbb{F}_2$, or, in general $\mathbb{F}_q$, $q = 2^m$
  - The network coding channel is an erasure channel, representing packet erasures due to errors in layers 1-3
  - Some assumptions made for physical layer coding do not hold (e.g. packets do not arrive in a synchronous fashion)

Classical scenarios

VSAT network(Two-Way Relay Channel)

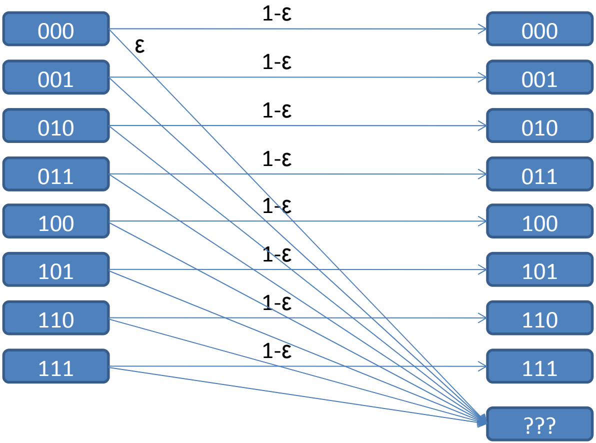


Butterfly network



Rateless Codes

- Rateless codes are upper layer codes for 1-hop broadcast scenario over erasure channels (or multiple hops but no intermediate processing)
- Example of 8-ary erasure channel



Rateless Codes

- Sources for PHY layer errors: decoding failures, link adaptation failures
- Strong PHY layer code behaves close to Shannon limit: either no errors or all bits in error
  - Errors easily detected by CRC or checksums, and packets dropped
  - At upper layers, link or end-to-end link may be seen as q-ary erasure channel, where $q = 2^m$ alphabet size
- Need for upper layer error correction

How to deal with erasures

Conventionally

- Use error control at link or transport layer based on retransmissions
- Requires feedback channel and typically inefficient:
  - Transmit ack, Transmit nack, Selective repeat or go back N,
  - If ϵ is large $\rightarrow$ throughput is small
- In principle, no need for retransmissions. Transmit at channel capacity $C_{\mathrm{RC}} = (1-\epsilon)L$ (bits/L-bit packet) and use powerful erasure correcting code

How to deal with erasures

For example Reed-Solomon codes

- Block code (N,K), alphabet size $q = 2^l$
- Property: the original K source symbols can be recovered as long as K out of the N transmitted symbols are received correctly
- Are optimal, but only practical for small K, N, q

$$\mathcal{O}(K(N-K)\mathrm{ld}N)$$

- Need to estimate ϵ to select the appropriate code rate $R = K/N$
- However, erasure rate often unknown or different for different users in a multicast

How to deal with erasures

Alternative: rateless codes or Fountain Codes

- Do not need to estimate the erasure rate
- Code rate can be determined and adapted on the fly
- Can potentially generate an unlimited number of encoded packets $\rightarrow$ rate granularity almost continuous
- Universal: nearly optimal for any erasure channel (does not depend on the channel statistics)
- Low encoding/decoding complexity

Fountain Codes

Encoding process

- Break down information message into K blocks/packets
- Transmit $N > K$ encoded blocks
- Fountain codes are characterized by the ability to produce a very large number of encoded packets from data bits

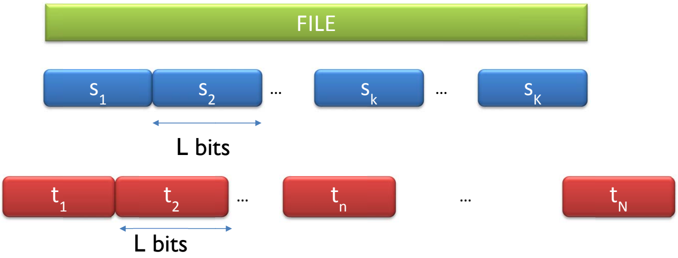
Decoding process

- Message can be decoded once $K = K + \epsilon$ packets have been received
- The order of the received packets is not important
- Code rate can be determined on the fly
- Stop transmission when ACK from all receivers on full message
- Data carousel (cyclic transmission of information, e.g. DVB)

Linear Random Coding

Encoding

- Consider an erasure channel over entire packets
- Message broken into K packets: $s_1, s_2,, s_K$



- For each transmitted packet t_n the encoder generates K random bits $\{G_{kn}\} \in \{0,1\}$ and performs modulo-2 sum of data packets for which $\{G_{kn}\} = 1$

$$t_n = \sum_{k=1}^{K} s_k G_{kn}$$

- Successful decoding of original data packets depends on the number of received encoded packets N
  - If $N < K \rightarrow$ cannot decode
  - If $N \geq K \rightarrow$ can decode if $\mathbf{G}$ is invertible module-2

$$\hat{s}_k = \sum_{n=1}^{N} t_n G_{nk}^{-1}$$

- The probability of correct decoding can be computed as the probability that $\mathbf{G}$ is invertible
- which is equivalent to the probability that each new column is linearly independent with the preceding ones

  Example: for $K > 10$

$$(1 - 2^{-K})(1 - 2^{-(K-1)}) \ldots (1 - 2^{-1}) = 0.289$$

- For $N = K + \epsilon > K$ with number of excess packets ϵ small the probability of matrix **G** containing an invertible $K \times K$ matrix, increases with ϵ as,

$$P \triangleq 1 - \delta(\epsilon)$$

with

$$\delta(\epsilon) \leq 2^{-\epsilon}$$

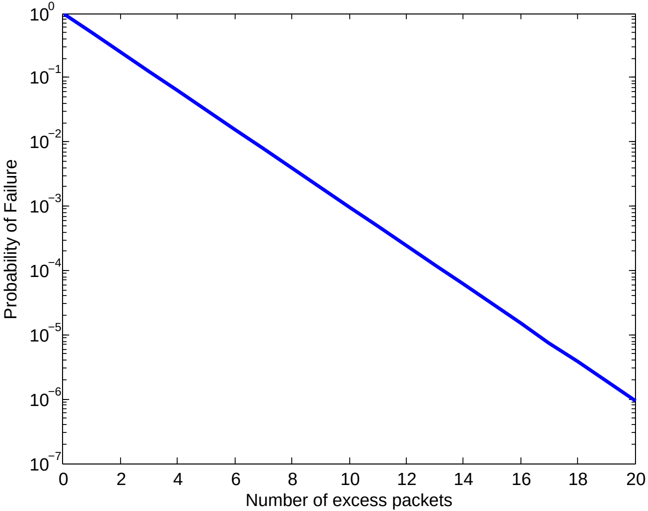
- I.e., the number of required packets to have $1 - \delta$ probability of success is approximately

$$K + \mathrm{ld}\frac{1}{\delta}$$

- The rate of Fountain Codes can be arbitrarily close to 1 for large K

Linear Random Coding

Upper Bound Probability of decoding failure



- Encoding: $\mathcal{O}(K/2)$ packet operations (modulo2 additions) on average
- Decoding:$\mathcal{O}(K^3 + K^2/2)$ matrix inversion and multiplication of coded packets
- Polynomial complexity is good compared to exponential complexity (e.g. random FEC or Reed-Solomon)
- Problem if we want high rates $\rightarrow K \upuparrows$
- Solution
  - LT Codes (Luby)
  - goal: retain performance of linear random coding at reduced complexity

- K source packets $s_1, s_2 \ldots, s_K$

  1. Choose packet degree d_n from a degree distribution
  2. Choose at random d_n distinct input packets from $\{s_k\}$ and set t_n equal to the bitwise modulo-2 packet addition

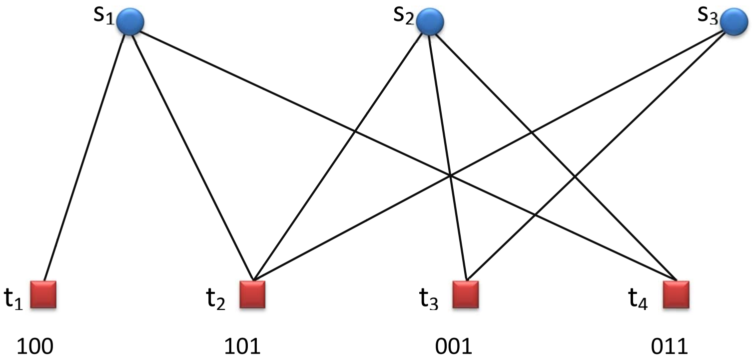
$$t_n = \sum_{k=1} d_n s_{i \sim \mathcal{U}(1,K)}$$

- Encoding process defines a graph connecting source packets to encoded packets with,
  - t_n check nodes
  - s_k variable nodes
  - If mean degree is $\ll K$ the graph is sparse $\rightarrow$ code has a low density generator matrix
- Decoding: message passing
- Message passing for erasure channels is simple: messages are either completely certain or completely uncertain

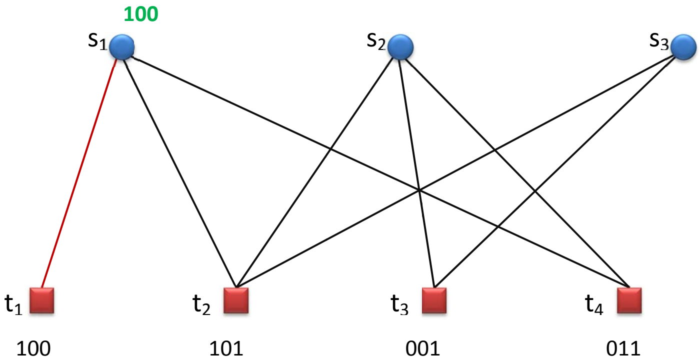
Algorithm

1. Find check node t_n connected to a single source packet s_k
2. Set $s_k = t_n$
3. Add s_k to all checks $t_{n'}$ connected to s_k, i.e., $\mathbf{G}_{n'k} = 1$
4. Remove edges connected to source packet s_k
5. Go back to Step 1 until all s_k are decoded

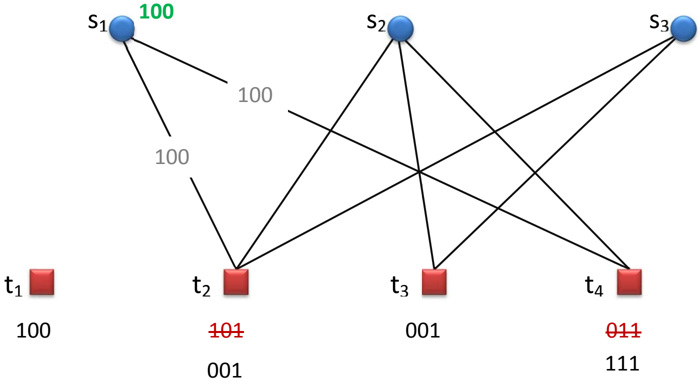
sparse graph



1. degree-one check node

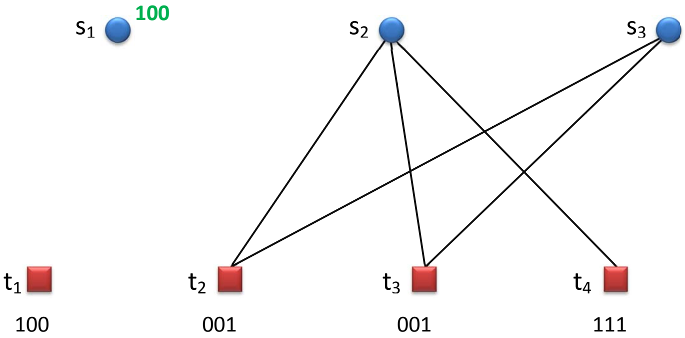


2. check-node update

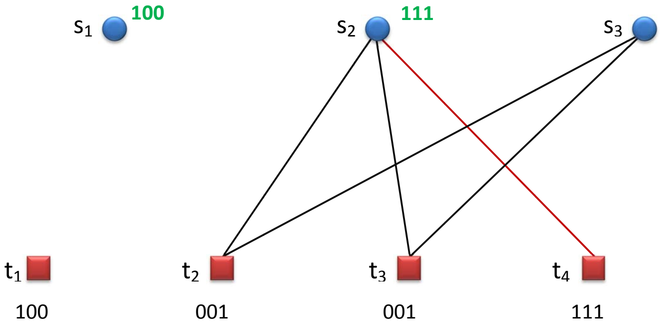


LT Codes
Example

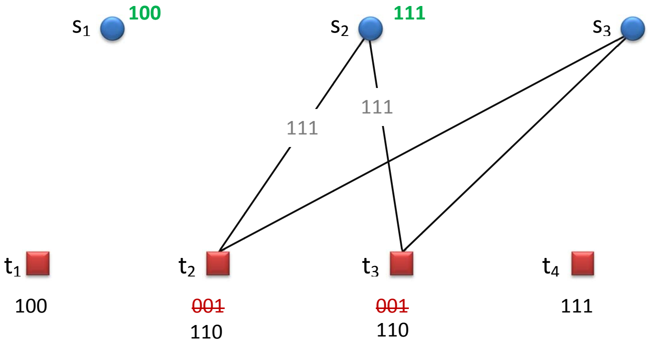
3. remove edges



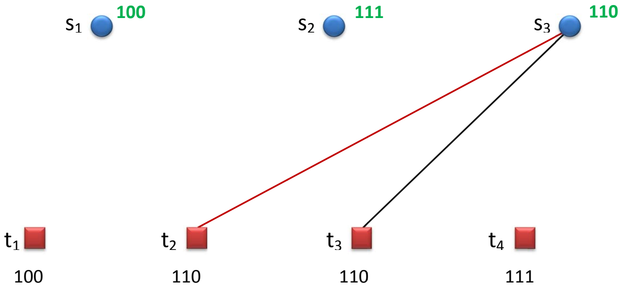
1. repeat the process: degree-one check node



2. repeat the process: check-node update



3. repeat the process: remove edges



- LT codes carefully design the degree distribution
  1. the decoding process does not get stuck
  2. the average node degree is small (sparse graph)
- Some packets must have high degree to ensure connectivity
- Majority of packets must have low degree to ensure the graph is sparse
- Tool density evolution
- Degree distribution robust soliton distribution
- Decoding complexity grows as $\mathcal{O}(K \ln K)$ as opposed to $\mathcal{O}(K^3 + K^2/2)$

Raptor Codes

- One step further
- Reduces both encoding and decoding to linear complexity
- How: concatenation of outer code (e.g. irregular LDPC) with a weakened LT code
  - LT code with very low average degree $\bar{d} = 3$
  - ensures the decoder does not get stuck
  - but a fraction of source packets are not connected to the graph $\rightarrow$ erased
  - Erasures are dealt with by the outer code (LPDC)

Applications

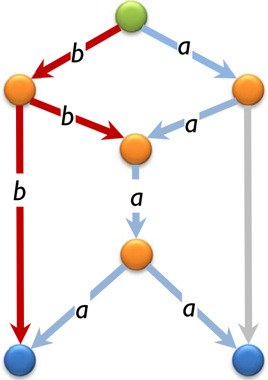
- Storage
  - Better protection against catastrophic disk failures than typical disk redundancy systems
  - Faster recovery in case of reading failures (no need to recover exactly that lost packet)
- Broadcast/Multicast
  - Avoid large amounts of retransmissions
  - Data carrousel approach:
    - users have opportunistic access to the channel and wish to download a fixed amount of data (e.g. road traffic info)
    - Encode data using an FC so that all users can decode regardless of when they connect to the channel
- Wireless Sensor Networks
  - Rateless codes adapted to WSN
  - Main application: data dissemination
  - Combines FC with opportunistic listening

- At the physical layer $\rightarrow$ link-level coding: combination of forward and feedback error correction (FEC + ARQ)
- At network level (end-to-end) $\rightarrow$ feedback encoding has prevailed so far
- Optimal in point-to-point links
- Complications arise when applied end-to-end
- Such feedback encoding mechanisms are difficult to implement as they sometimes need to deal with other problems, mainly network congestion (e.g. TCP congestion control mechanism)
- Feedback encoding may be undesirable in terms of delay if the end-to-end path is long
- It may also be undesirable for multicast connections

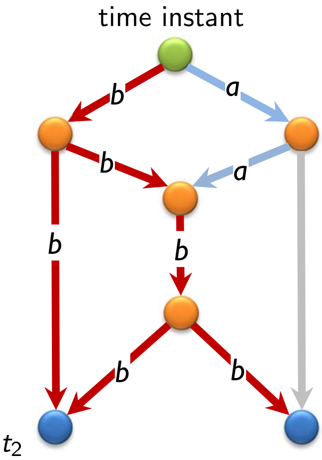
Feedforward network encoding

- Intermediate nodes store packets in memory
- They retransmit a linear transformation of packets in memory
- Properties:
  - Encoding performed at packet level
  - May approach capacity
  - Can be operated ratelessly
  - Polynomial time decoding (as in Fountain Codes)

- Network composed by one source, two sinks, unit-capacity edges
- Can route two packets a,b to two sinks with time sharing
  1. Time instant t_1: sink 2 receives a and b; sink 1 receives a
  2. Time instant t_2: sink 1 receives a and b; sink 2 receives a
- Multicast rate of 1.5 packets per use of the network

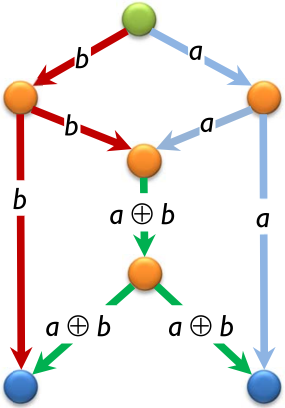


- Network composed by one source, two sinks, unit-capacity edges
- Can route two packets a, b to two sinks with time sharing
  1. Time instant t_1: sink 2 receives a and b; sink 1 receives a
  2. Time instant t_2: sink 1 receives a and b; sink 2 receives a
- Multicast rate of 1.5 packets per use of the network



- Network composed by one source, two sinks, unit-capacity edges
- Can route two packets a,b to two sinks with network coding
    1. Time instant t_1: sink 2 receives a and $a \oplus b$; sink 1 receives b, $a \oplus b$
    2. Sink 1 receives b, can recover a from $a \oplus b$: $(a \oplus b) \oplus b = a$
    3. Sink 2 receives a, can recover b from $a \oplus b$: $(a \oplus b) \oplus a = b$
- Multicast rate of 2 packets per use of the network

time instant t_1



- Random linear network coding (distributed scheme) achieves multicast capacity as long as the Galois Field order q is sufficiently large
- Generates random linear combinations of incoming packets

  $u_k \in \mathbb{F}_q^N$, $y_n = \sum_{i=1}^{K} G_{ni} u_i$, $G_{ni} \in \mathbb{F}_q^N$
- y_n innovative packets
- $\mathbf{g}_i = (G_{1i}, \ldots, G_{Ni})$ are global encoding vectors

- Sink nodes perform Gaussian elimination on the set of global encoding vectors of packets in its memory

$$\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} G_{11} & \dots & G_{1K} \\ \vdots & \ddots & \vdots \\ G_{N1} & \dots & G_{NK} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_K \end{bmatrix}$$

- If matrix $\mathbf{G}$ has rank K, packets $u_1, u_2, \dots, u_K$ are recovered
- Global encoding vectors must be known by receiver
- It can be included in packet as side information (e.g. header)
- Rate loss is minor for large packets

- Each arriving packet represents a linear constraint of the form

$$y = \mathbf{g}\,[u_1, \ldots, u_K]$$

- restricts each component of source vectors to $(K-1)$-dimensional subspace
- Innovative packets $\triangleq$ vector $\mathbf{g}$ outside subspace spanned by vectors $\mathbf{g}$ of packets in buffer
- Non-innovative packets $\triangleq$ vector $\mathbf{g}$ lies inside that subspace
- Non-innovative packets are not useful to produce linear combinations of outgoing packets and can be discarded
- This limits node buffering capability to K packets at most

- Non-innovative packets do not preclude correct decoding but are useless, and their transmission should be avoided
- Rate control
    - Transmission rate goes to zero if nodes continue to retransmit linear combinations of packets in their buffer
    - Possible solution is to define *generations* of information packets, and stop transmitting packets of one generation when first packet from new generation arrives

References - Coded Cooperation

HunNos04 T. Hunter and A. Nosratinia, Performance analysis of coded cooperation, in Proc. of IEEE International Conference on Communications (ICC), Anchorage, Alaska, May 2003, pp. 2688-2692, Vol. 4.

HunNos06 T.E. Hunter, S. Sanayei and A.Nosratinia,Outage Analysis of Coded Cooperation, in IEEE Transactions on Information Theory, vol. 52, no. 2, pp. 375-391, Feb. 2006.

SteErk A. Stefanov and E. Erkip, Cooperative Coding for Wireless Networks, in IEEE Transactions on Communications, vol. 52, no. 9, pp. 1470-1476, Sep. 2004.

ValZha03 M.C. Valenti and B. Zhao, Distributed turbo codes: Towards the capacity of the relay channel, in Proc. IEEE Vehicular Tech. Conf. (VTC), (Orlando, FL), Oct. 2003.

LiVuc06 Y. Li et al., Distributed Turbo Coding with Soft Information Relaying in Multihop Relay Networks, IEEE JSAC, vol. 24, no. 11, Nov. 2006, pp. 204050.

ChaBay07 A. Chakrabarti, A. de Baynast, A. Sabharwal and B. Aazhang, Low Density Parity Check Codes for the Relay Channel, in IEEE JSAC, vol. 25, no. 2, pp. 280-291, Feb. 2007.