An Algebraic Approach to Network Coding

July 30–31, 2009

Outline

Digital Communication

Digital communication networks are integral parts of our lives these days; so, we want to know how to most effectively use these networks.

Digital Communication

Digital communication networks are integral parts of our lives these days; so, we want to know how to most effectively use these networks.

Two questions quickly arise when considering how to do this:

- What if an error occurs when transmitting data?

Digital Communication

Digital communication networks are integral parts of our lives these days; so, we want to know how to most effectively use these networks.

Two questions quickly arise when considering how to do this:

- What if an error occurs when transmitting data?
- How do we send as much data as possible through the network.

Digital Communication

Digital communication networks are integral parts of our lives these days; so, we want to know how to most effectively use these networks.

Two questions quickly arise when considering how to do this:

- What if an error occurs when transmitting data?
- How do we send as much data as possible through the network.

We will be focusing on the latter of these two questions.

Network Coding

Traditionally, the method of choice for transmitting data at a node has been routing. In this method, data recieved at a node is simply re-transmitted (selectively) by the node.

Network Coding

Traditionally, the method of choice for transmitting data at a node has been routing. In this method, data recieved at a node is simply re-transmitted (selectively) by the node.

Network coding is a simple extension of routing that allows for any node in the network to perform operations on its recieved data before it transmits any data.

Network Coding

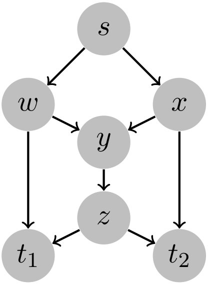
Traditionally, the method of choice for transmitting data at a node has been routing. In this method, data recieved at a node is simply re-transmitted (selectively) by the node.

Network coding is a simple extension of routing that allows for any node in the network to perform operations on its recieved data before it transmits any data.

Throughout this discussion, we will be considering "delay-free" networks.

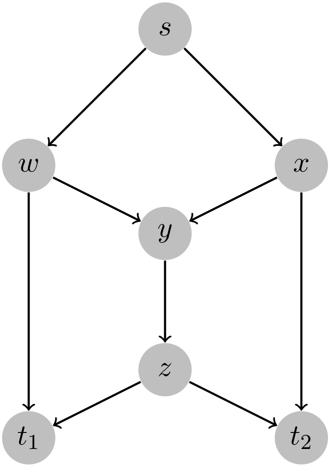
An Example: The Butterfly Network

In the network below, we want to get two separate messages from the source node s to each of the sink nodes t_1 and t_2.

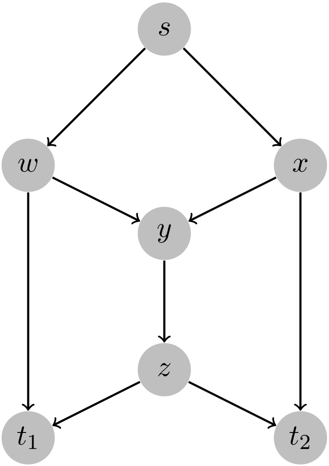


An Example: The Butterfly Network

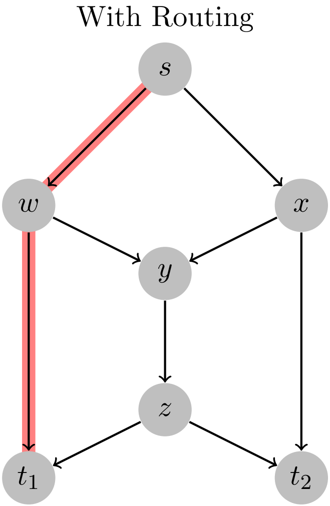
With Routing

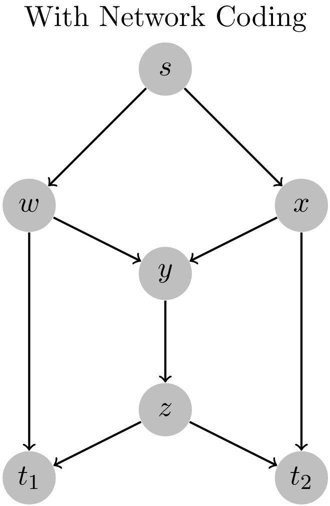


With Network Coding

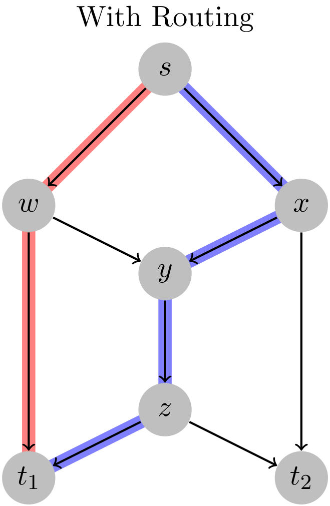


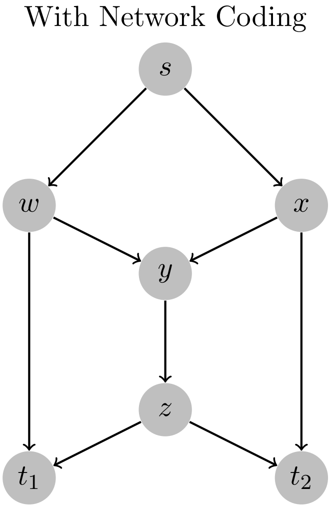
An Example: The Butterfly Network





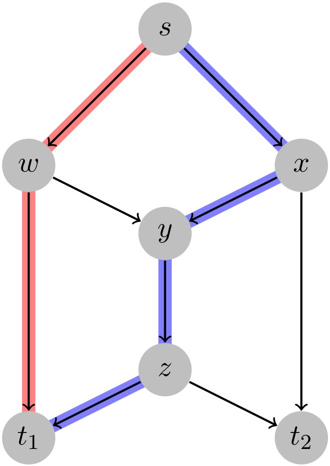
An Example: The Butterfly Network



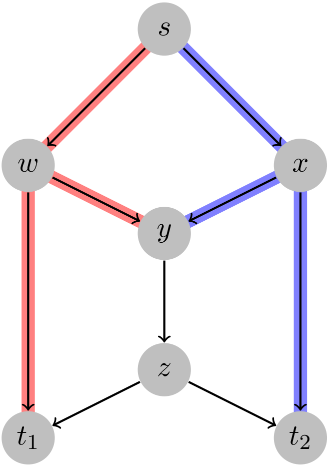


An Example: The Butterfly Network

With Routing

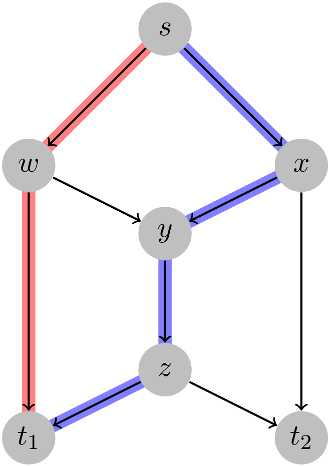


With Network Coding

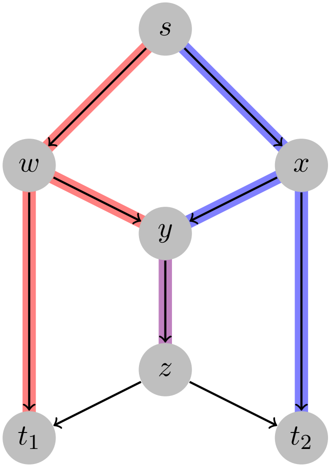


An Example: The Butterfly Network

With Routing



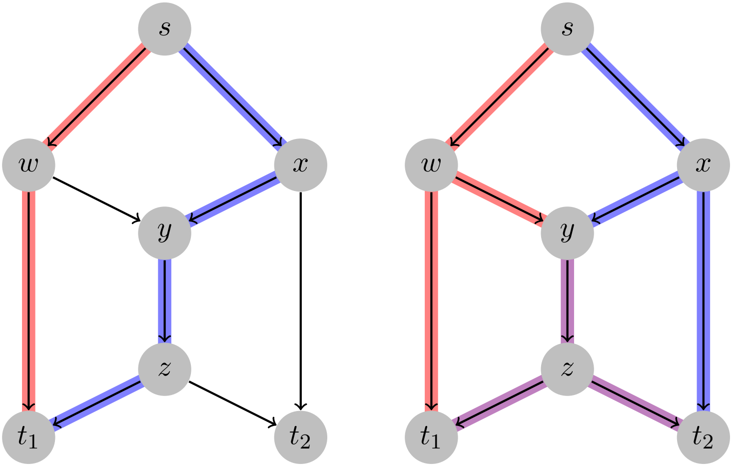
With Network Coding



An Example: The Butterfly Network

With Routing

With Network Coding



Outline

Some Definitions

For the purposes of this presentation, we will define a communication network to be a directed graph.

Some Definitions

For the purposes of this presentation, we will define a communication network to be a directed graph.

Definition

- A directed graph $\mathcal{G}$ is an ordered pair (V, E) where V is a collection of points and E is a subset of $V \times V \times \mathbb{N}$.

Some Definitions

For the purposes of this presentation, we will define a communication network to be a directed graph.

Definition

- A directed graph $\mathcal{G}$ is an ordered pair (V, E) where V is a collection of points and E is a subset of $V \times V \times \mathbb{N}$.
- The set V is refered to as the vertex set of $\mathcal{G}$, and the set E is refered to as the edge set of $\mathcal{G}$.

Some Definitions

For the purposes of this presentation, we will define a communication network to be a directed graph.

Definition

- A directed graph $\mathcal{G}$ is an ordered pair (V, E) where V is a collection of points and E is a subset of $V \times V \times \mathbb{N}$.
- The set V is refered to as the vertex set of $\mathcal{G}$, and the set E is refered to as the edge set of $\mathcal{G}$.
- An element of V is called a vertex (or a node), and an element of E is called an edge (or a link).

Some Definitions

For the purposes of this presentation, we will define a communication network to be a directed graph.

Definition

- A directed graph $\mathcal{G}$ is an ordered pair (V, E) where V is a collection of points and E is a subset of $V \times V \times \mathbb{N}$.
- The set V is refered to as the vertex set of $\mathcal{G}$, and the set E is refered to as the edge set of $\mathcal{G}$.
- An element of V is called a vertex (or a node), and an element of E is called an edge (or a link).

Note: The set E is a subset of $V \times V \times \mathbb{N}$ to allow for multiple edges between any two vertices. When no confusion can arise, we will denote an edge as an ordered pair in $V \times V$.

A Simple Example

Let $\mathcal{G} = (V, E)$ be a directed graph with V and E given as follows:

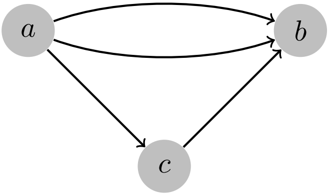
$$V = \{a, b, c\}$$
$$E = \{(a, b, 1), (a, b, 2), (a, c, 1), (c, b, 1)\}$$

A Simple Example

Let $\mathcal{G} = (V, E)$ be a directed graph with V and E given as follows:

$$V = \{a, b, c\}$$
$$E = \{(a, b, 1), (a, b, 2), (a, c, 1), (c, b, 1)\}$$

Then we can visualize $\mathcal{G}$ as the following figure:

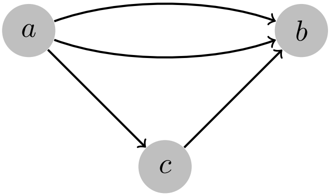


A Simple Example

Let $\mathcal{G} = (V, E)$ be a directed graph with V and E given as follows:

$$V = \{a, b, c\}$$
$$E = \{(a, b, 1), (a, b, 2), (a, c), (c, b)\}$$

Then we can visualize $\mathcal{G}$ as the following figure:



Some Notation

- **From here on out, $\mathcal{G} = (V, E)$ will represent a directed graph.**

Some Notation

- **From here on out, $\mathcal{G} = (V, E)$ will represent a directed graph.**
- For an edge $e = (v, u, i) \in E$, we call u and v the head and tail of e, respectively, and we use the notation $\text{head}(e) = u$ and $\text{tail}(e) = v$.

Some Notation

- **From here on out, $\mathcal{G} = (V, E)$ will represent a directed graph.**
- For an edge $e = (v, u, i) \in E$, we call u and v the head and tail of e, respectively, and we use the notation $\text{head}(e) = u$ and $\text{tail}(e) = v$.
- For a vertex $v \in V$, we define the sets

$$\Gamma_I(v) := \{e \in E \mid \text{head}(e) = v\} \qquad \text{and}$$
$$\Gamma_O(v) := \{e \in E \mid \text{tail}(e) = v\}.$$

We define the in-degree and out-degree of v to be $\delta_I(v) := |\Gamma_I(v)|$ and $\delta_O(v) := |\Gamma_O(v)|$, respectively.

Modeling Data

We will model the data transmitted through a network as a collection of random processes.

Modeling Data

We will model the data transmitted through a network as a collection of random processes.

- For each $v \in V$, we let $\mathcal{X}(v) = \{X(v, 1), \ldots, X\big(v, \mu(v)\big)\}$ denote a collection of random processes that are observable at the node v. Here, $\mu(v)$ is allowed to equal zero, in which case the collection $\mathcal{X}(v)$ is empty.

Modeling Data

We will model the data transmitted through a network as a collection of random processes.

- For each $v \in V$, we let $\mathcal{X}(v) = \{X(v,1), \ldots, X\big(v, \mu(v)\big)\}$ denote a collection of random processes that are observable at the node v. Here, $\mu(v)$ is allowed to equal zero, in which case the collection $\mathcal{X}(v)$ is empty.
- With each edge $e \in E$ we will associate a random process $Y(e)$. This random process models the data transmitted over the link e.

Transmitting Data

We will assume that communication in our network is performed by transmission of vectors of bits, that is, as elements of $\{0, 1\}^m$ for some $m \in \mathbb{N}$. Unless otherwise stated, a vector is assumed to be a row vector.

Transmitting Data

We will assume that communication in our network is performed by transmission of vectors of bits, that is, as elements of $\{0, 1\}^m$ for some $m \in \mathbb{N}$. Unless otherwise stated, a vector is assumed to be a row vector.

The above assumption allows for the following:

- We can interpret any random process R as a discrete random process $R = \{R_0, R_1, R_2, \dots\}$.

Transmitting Data

We will assume that communication in our network is performed by transmission of vectors of bits, that is, as elements of $\{0,1\}^m$ for some $m \in \mathbb{N}$. Unless otherwise stated, a vector is assumed to be a row vector.

The above assumption allows for the following:

- We can interpret any random process R as a discrete random process $R = \{R_0, R_1, R_2, \dots\}$.
- The data that we are working with, binary m-tuples, can now be interpreted as elements of $\mathbb{F}_{2^m}$. This has the benefit of using the operations of $\mathbb{F}_{2^m}$ on our "packets" of data.

Will the definitions ever end?

Definition

- A cycle in $\mathcal{G}$ is a sequence of edges of the form $\{(v_0, v_1), (v_1, v_2), \ldots, (v_n, v_0)\}$.

Will the definitions ever end?

Definition

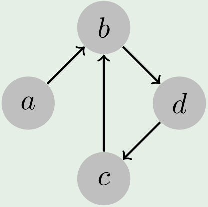
- A cycle in $\mathcal{G}$ is a sequence of edges of the form $\{(v_0, v_1), (v_1, v_2), \ldots, (v_n, v_0)\}$.
- The graph $\mathcal{G}$ is called acyclic if it contains no cycles.

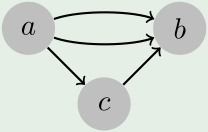
Will the definitions ever end?

Definition

- A cycle in $\mathcal{G}$ is a sequence of edges of the form $\{(v_0, v_1), (v_1, v_2), \ldots, (v_n, v_0)\}$.
- The graph $\mathcal{G}$ is called acyclic if it contains no cycles.

Example





They have to end eventually...

Definition

- A connection in $\mathcal{G}$ is a triple $(v, v', \mathcal{X}(v, v')) \in V \times V \times \mathcal{P}_{\mathcal{X}(v)}$, where $\mathcal{P}_{\mathcal{X}(v)}$ denotes the power set of $\mathcal{X}(v)$.

They have to end eventually...

Definition

- A connection in $\mathcal{G}$ is a triple $(v, v', \mathcal{X}(v, v')) \in V \times V \times \mathcal{P}_{\mathcal{X}(v)}$, where $\mathcal{P}_{\mathcal{X}(v)}$ denotes the power set of $\mathcal{X}(v)$.
- Given a connection $c = (v, v', \mathcal{X}(v, v'))$, we call v the source of c and v' the sink of c. We will denote the source and sink of c by $\text{source}(c)$ and $\text{sink}(c)$, respectively.

They have to end eventually...

Definition

- A connection in $\mathcal{G}$ is a triple $\big(v, v', \mathcal{X}(v, v')\big) \in V \times V \times \mathcal{P}_{\mathcal{X}(v)}$, where $\mathcal{P}_{\mathcal{X}(v)}$ denotes the power set of $\mathcal{X}(v)$.
- Given a connection $c = \big(v, v', \mathcal{X}(v, v')\big)$, we call v the source of c and v' the sink of c. We will denote the source and sink of c by $\text{source}(c)$ and $\text{sink}(c)$, respectively.

If v is a sink of any connection c, the collection of random processes $\mathcal{Z}(v) = \{Z(v, 1), \ldots, Z\big(v, \nu(v)\big)\}$ denotes the output at v. These random processes will be functions of the random processes on the edges in $\Gamma_I(v)$.

Outline

$\mathbb{F}_{2^m}$-Linear Networks

Definition

Let $\mathcal{G} = (V, e)$ be a delay-free communication network. We say that $\mathcal{G}$ is an $\mathbb{F}_{2^m}$-linear network if for all links $e = (v, u, i) \in E$ the random process $Y(e)$ on e satisfies

$$Y(e) = \sum_{k=1}^{\mu(v)} \alpha_{k,e} X(v,k) + \sum_{e':\text{head}(e')=\text{tail}(e)} \beta_{e',e} Y(e')$$

where the coefficients $\alpha_{k,e}$ and $\beta_{e',e}$ are elements of $\mathbb{F}_{2^m}$.

Output of a Connection

In the case of a linear network, we will be forming the output processes in $\mathcal{Z}(v)$ as linear combinations of the edges in $\Gamma_I(v)$:

$$Z(v,k) = \sum_{e':\text{head}(e')=v} \varepsilon_{e',k} Y(e').$$

Output of a Connection

In the case of a linear network, we will be forming the output processes in $\mathcal{Z}(v)$ as linear combinations of the edges in $\Gamma_I(v)$:

$$Z(v,k) = \sum_{e':\text{head}(e')=v} \varepsilon_{e',k} Y(e').$$

Because all of the operations performed at every node are linear functions, each of the output processes is a linear combination of the input processes. Letting $\underline{x}$ denote the vector of input processes and $\underline{z}$ denote the vector of output processes, we find that there is a matrix M, referred to as a <u>transfer matrix</u>, such that $\underline{z} = \underline{x}M$.

Another Definition

Definition

We define a network coding problem as a pair $(\mathcal{G}, \mathcal{C})$ where $\mathcal{G}$ is a network and $\mathcal{C}$ is a set of desired connections.

Another Definition

Definition

We define a network coding problem as a pair $(\mathcal{G}, \mathcal{C})$ where $\mathcal{G}$ is a network and $\mathcal{C}$ is a set of desired connections.

We wish to give succinct algebraic conditions under which a set of desired connections is feasible. This is equivalent to finding elements $\alpha_{k,e}$, $\beta_{e',e'}$, and $\varepsilon_{e',j}$ in a suitably chosen field $\mathbb{F}_{2^n}$ such that all desired connections can be established. If a connection is feasible, we call the associated network coding problem solvable and the associated collection of coefficients a solution.

Outline

Tomorrow

Now that we have a basic framework set up for studying networks, we aim to answer some questions in the upcoming presentation:

Tomorrow

Now that we have a basic framework set up for studying networks, we aim to answer some questions in the upcoming presentation:

- Under what conditions is a given linear network coding problem solvable?

Tomorrow

Now that we have a basic framework set up for studying networks, we aim to answer some questions in the upcoming presentation:

- Under what conditions is a given linear network coding problem solvable?
- How can we efficiently find a solution to a given linear network coding problem?

Tomorrow

Now that we have a basic framework set up for studying networks, we aim to answer some questions in the upcoming presentation:

- Under what conditions is a given linear network coding problem solvable?
- How can we efficiently find a solution to a given linear network coding problem?
- When does a static solution exist for a network that is subject to link failures?