Performance improvements to peer-to-peer file transfers using network coding

Aaron Kelley
April 29, 2009
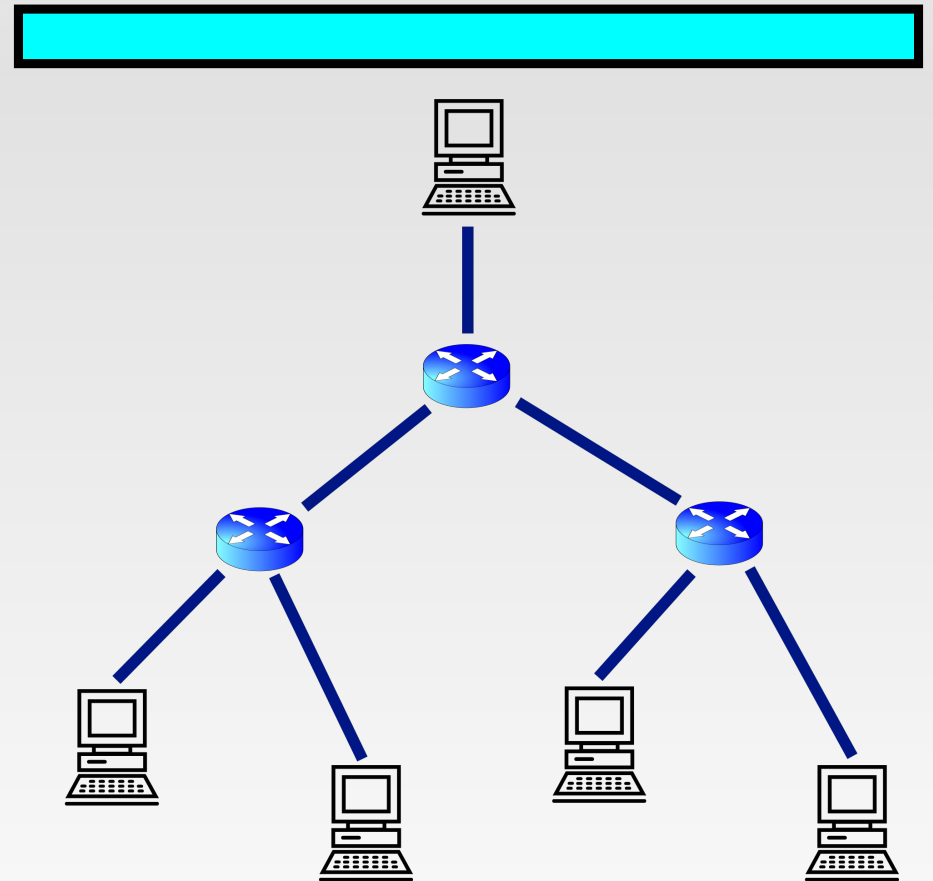
Mentor:
Dr. David Sturgill

Outline

- Introduction

- Network Coding Background

- Contributions

  - Precomputation in GF8

  - Batch encode via matrix multiplication
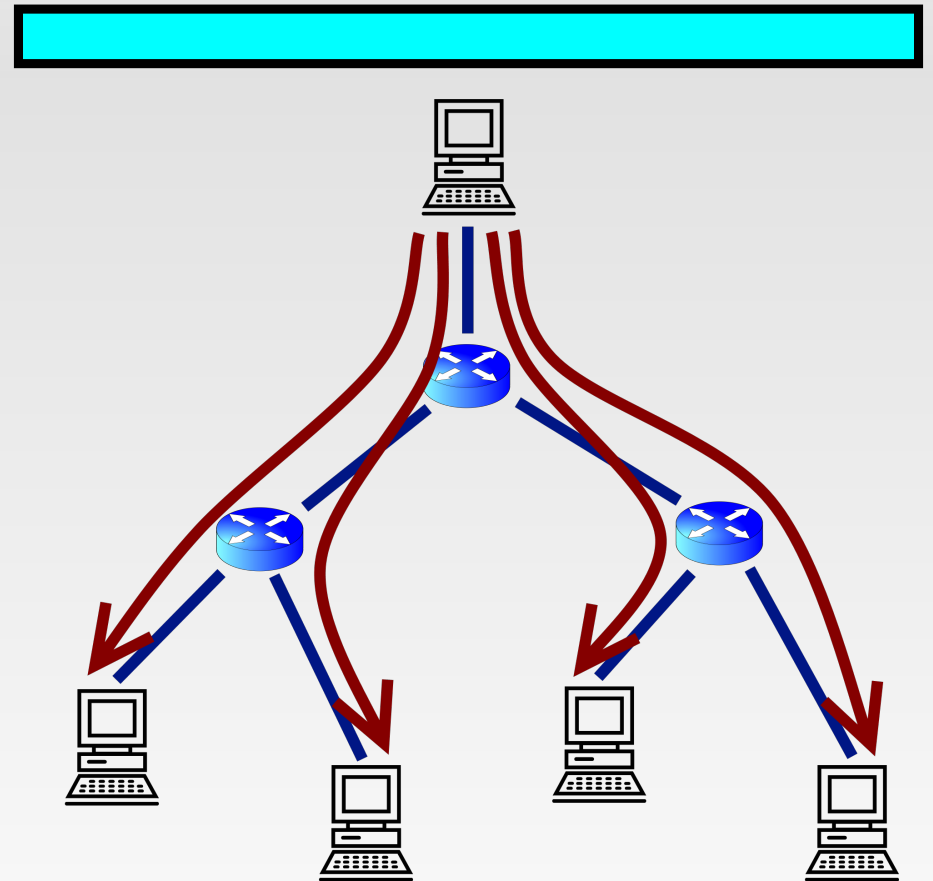
- Results

- Questions

Introduction
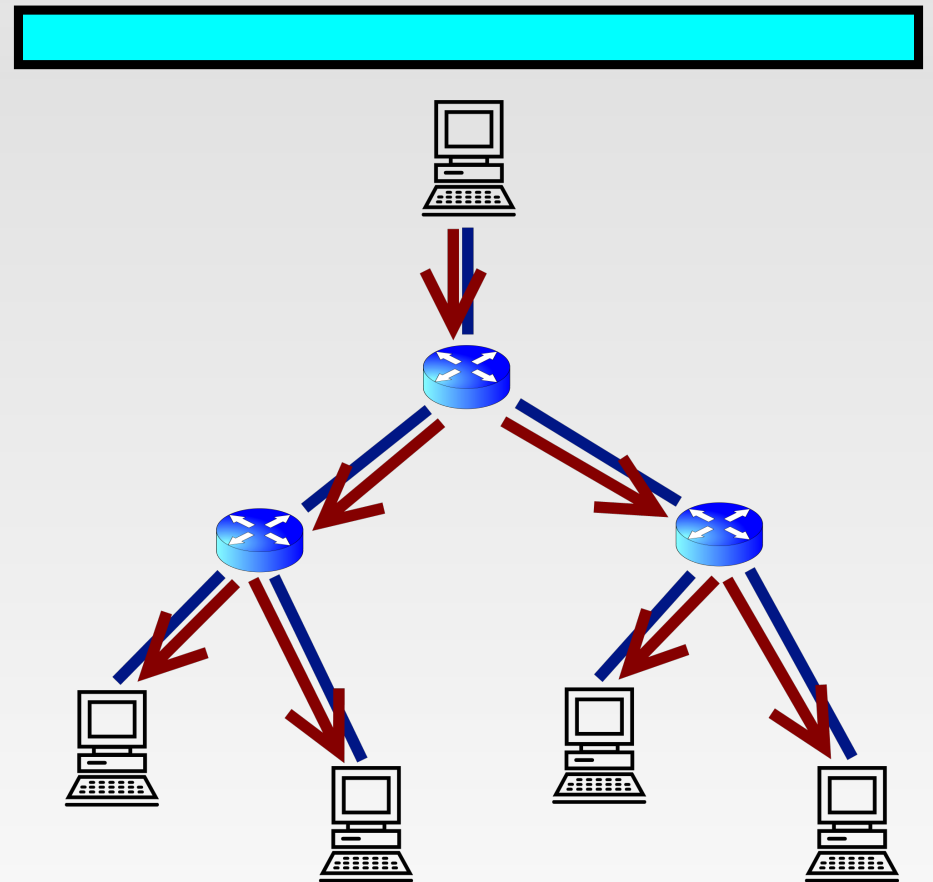
- Large file distribution – how do we do it?

Introduction

- Large file distribution – how do we do it?
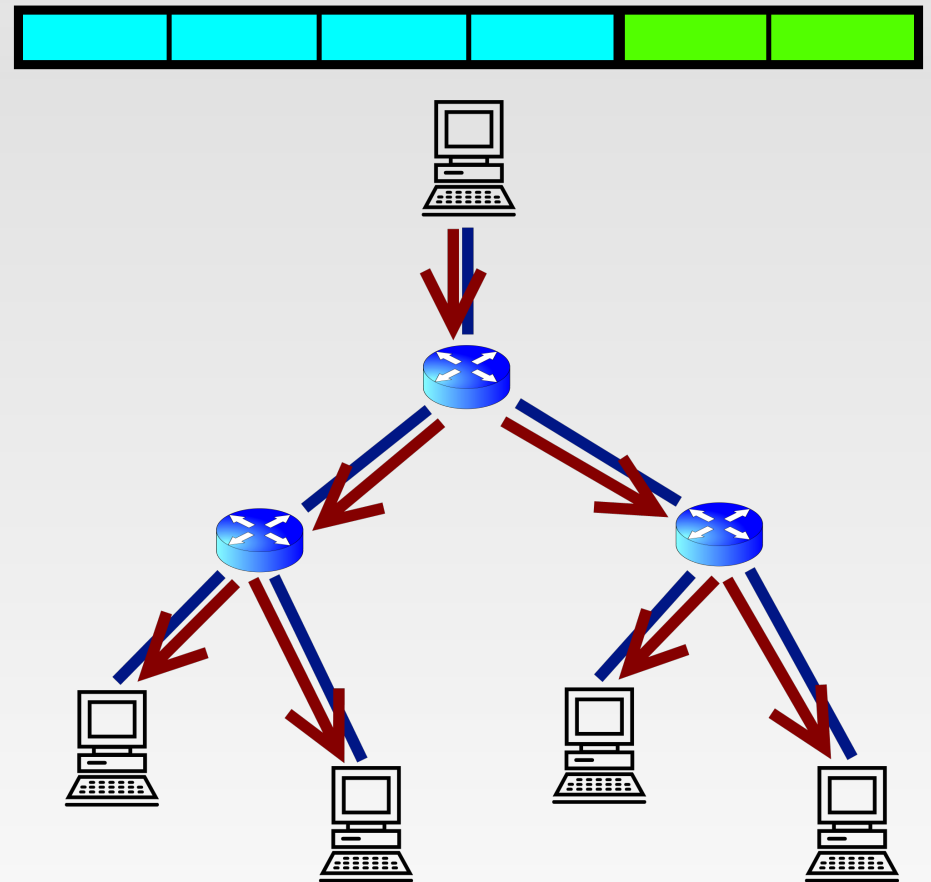 - HTTP / Unicast

Introduction

- Large file distribution – how do we do it?
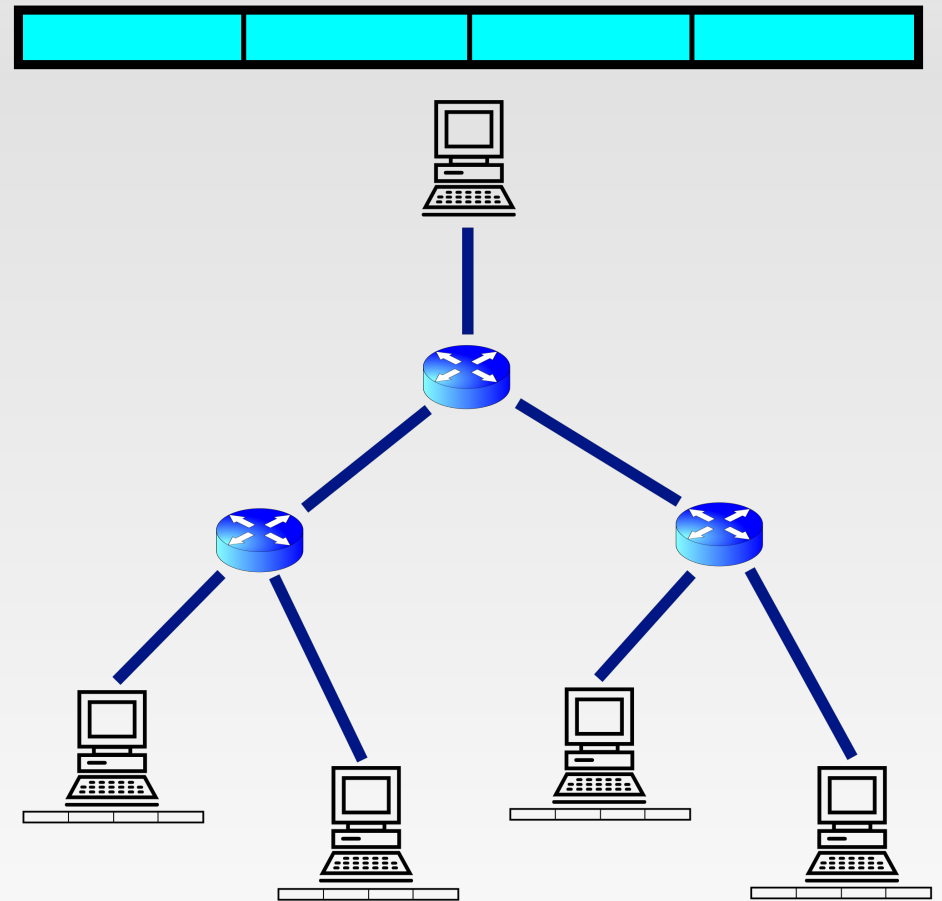 - HTTP / Unicast
  - Multicast

Introduction

- Large file distribution – how do we do it?
  - HTTP / Unicast
  - Multicast
  - Erasure codes

Introduction

- ## Large file distribution – how do we do it?

  - HTTP / Unicast

  - Multicast

  - Erasure codes

  - BitTorrent / Peer-to-peer

Introduction

- Large file distribution – how do we do it?
  - HTTP / Unicast
  - Multicast
  - Erasure codes
  - BitTorrent / Peer-to-peer

Introduction

- Large file distribution – how do we do it?
    - HTTP / Unicast
    - Multicast
    - Erasure codes
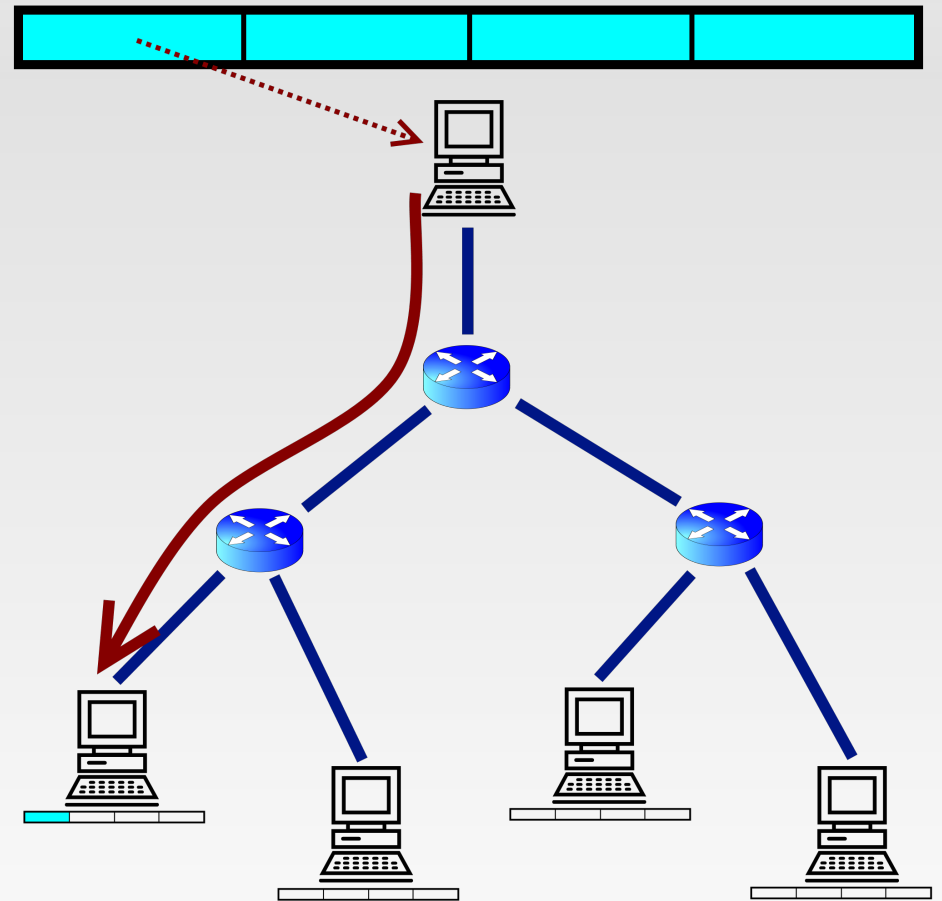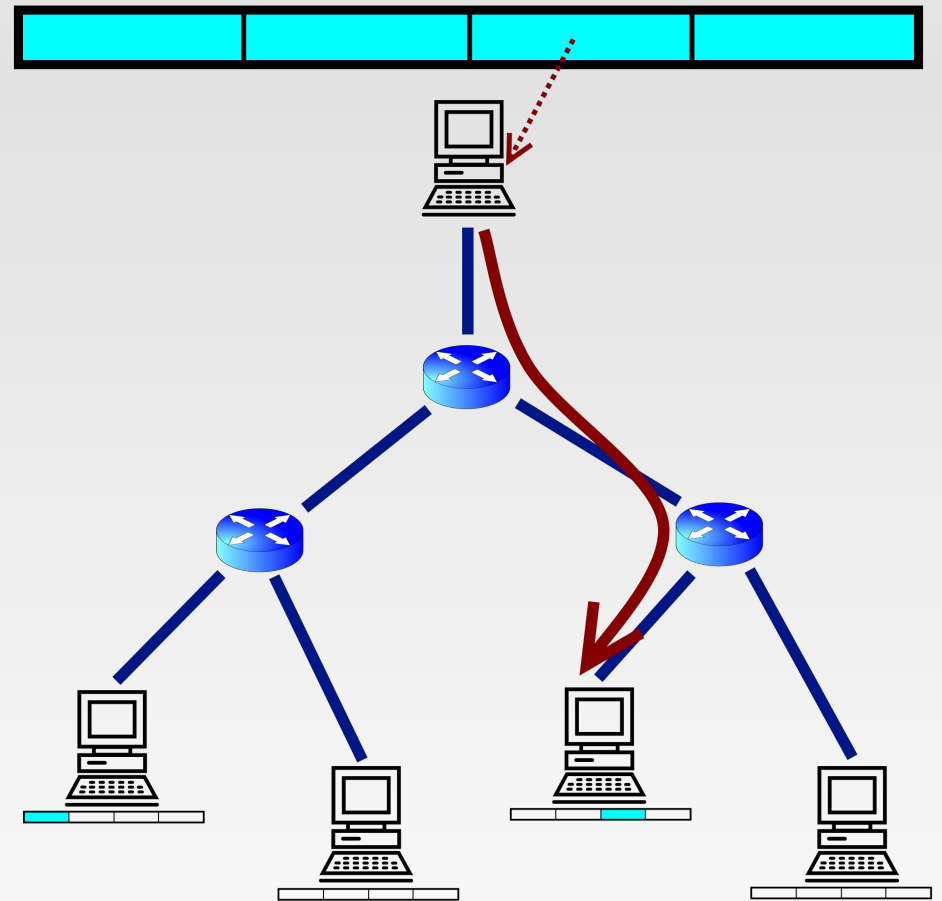   - BitTorrent / Peer-to-peer

Introduction

- Large file distribution – how do we do it?
  - HTTP / Unicast
  - Multicast
  - Erasure codes
  - BitTorrent / Peer-to-peer

Introduction

- Large file distribution – how do we do it?

  - HTTP / Unicast

  - Multicast

  - Erasure codes
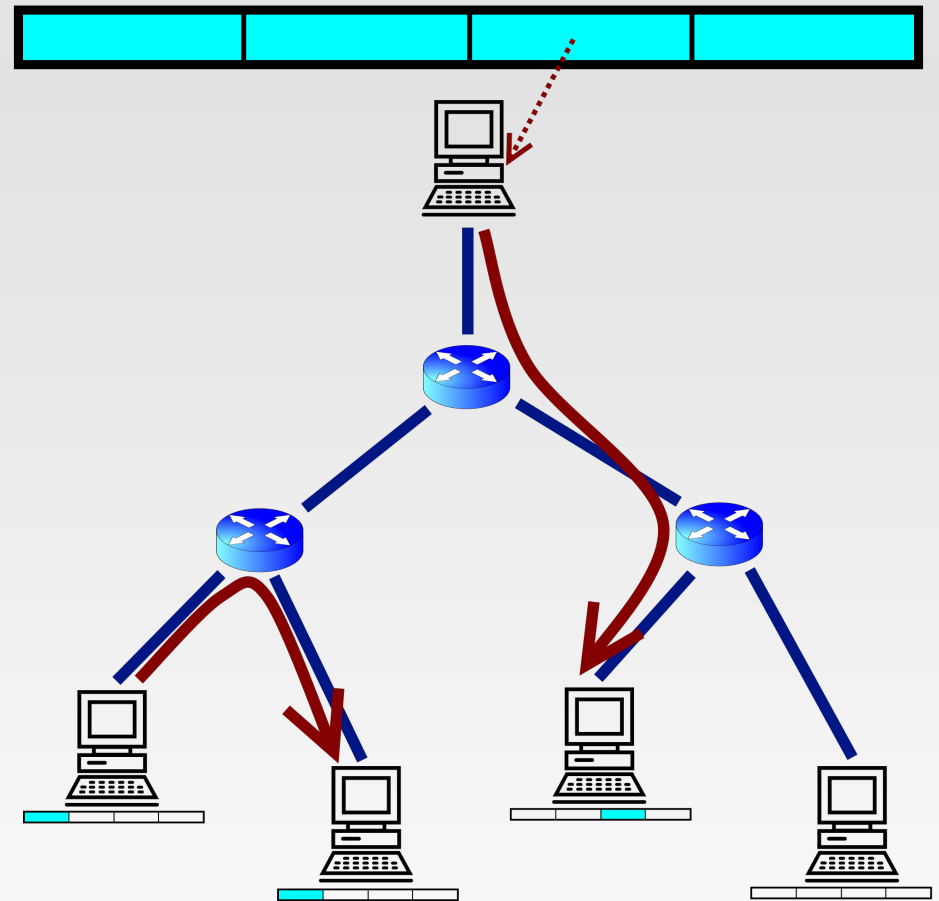
  - BitTorrent / Peer-to-peer

Problems with BitTorrent

- Peers must keep track of who has which pieces of the file.

- Duplicate data is still sent over network links – potentially wasteful in some situations.

Network Coding

- File is split into blocks, as with BitTorrent.

| 57 | 83 | 99 | 17 |
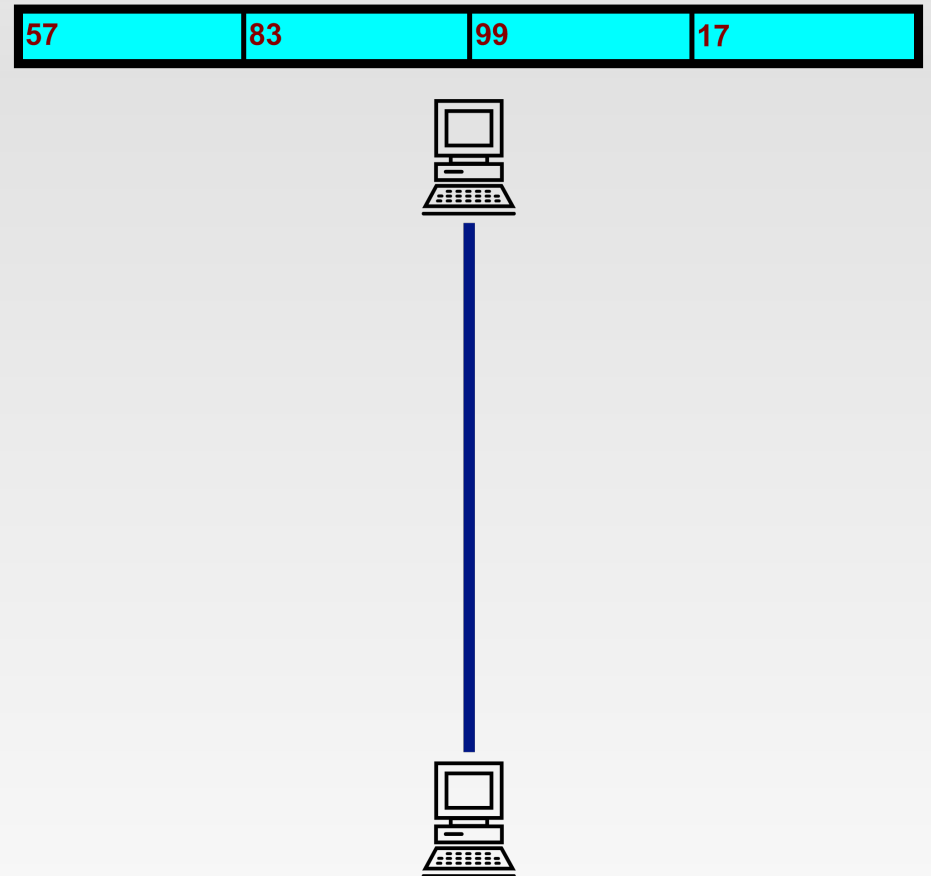
Network Coding

- File is split into blocks, as with BitTorrent.

- When preparing to send a block, source chooses n random coefficients.
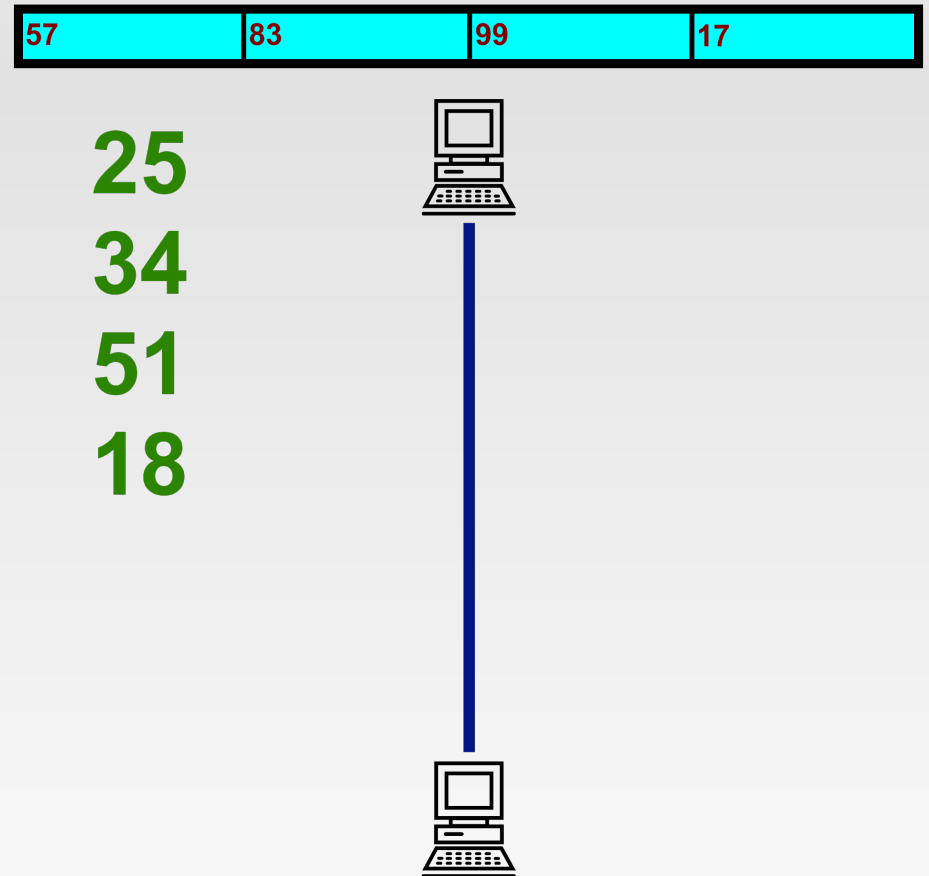
57	83	99	17

25
34
51
18

Network Coding

- Transmits a linear combination of all of the blocks in the file, along with these coefficients.

Network Coding

- Transmits a linear combination of all of the blocks in the file, along with these coefficients.

Network Coding

- Transmits a linear combination of all of the blocks in the file, along with these coefficients.

| 57 | 83 | 99 | 17 |

25 34 51 18   9602
66 71 81 35 18269
77 90 11 13 13169
51 88 89 45 19787

Network Coding

- Transmits a linear combination of all of the blocks in the file, along with these coefficients.

- After receiving *n* of these, a peer can decode the original file.

57	83	99	17

25 34 51 18   9602
66 71 81 35   18269
77 90 11 13   13169
51 88 89 45   19787
57 83 99 17

Network Coding

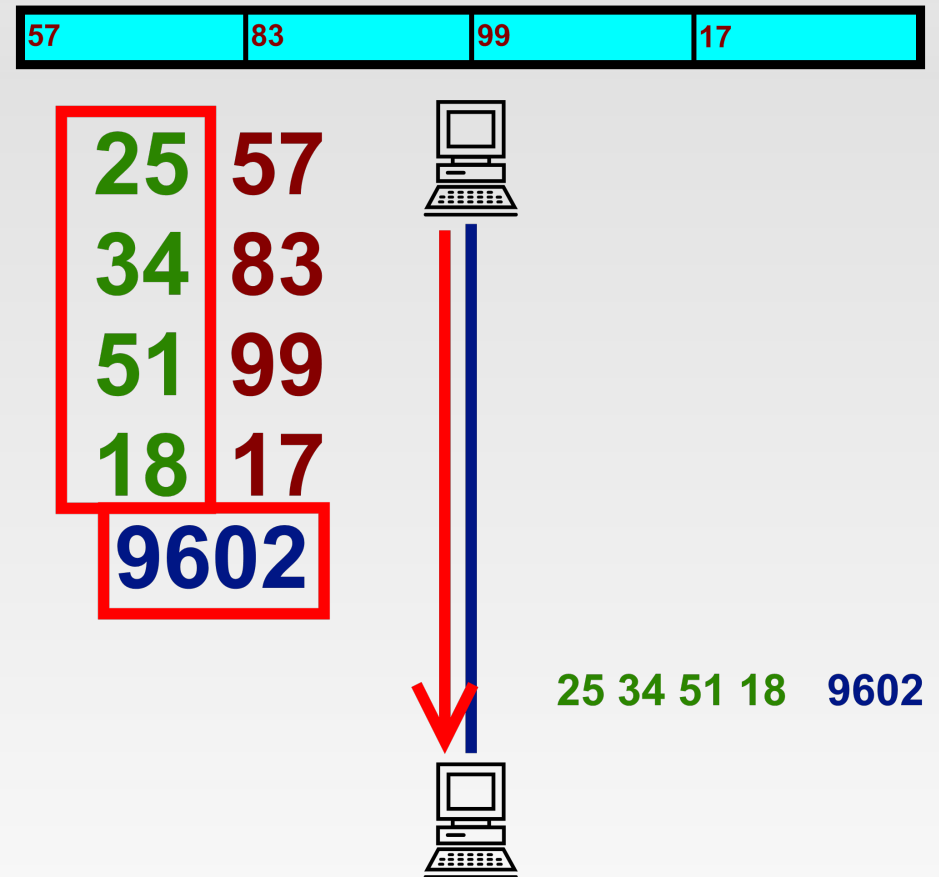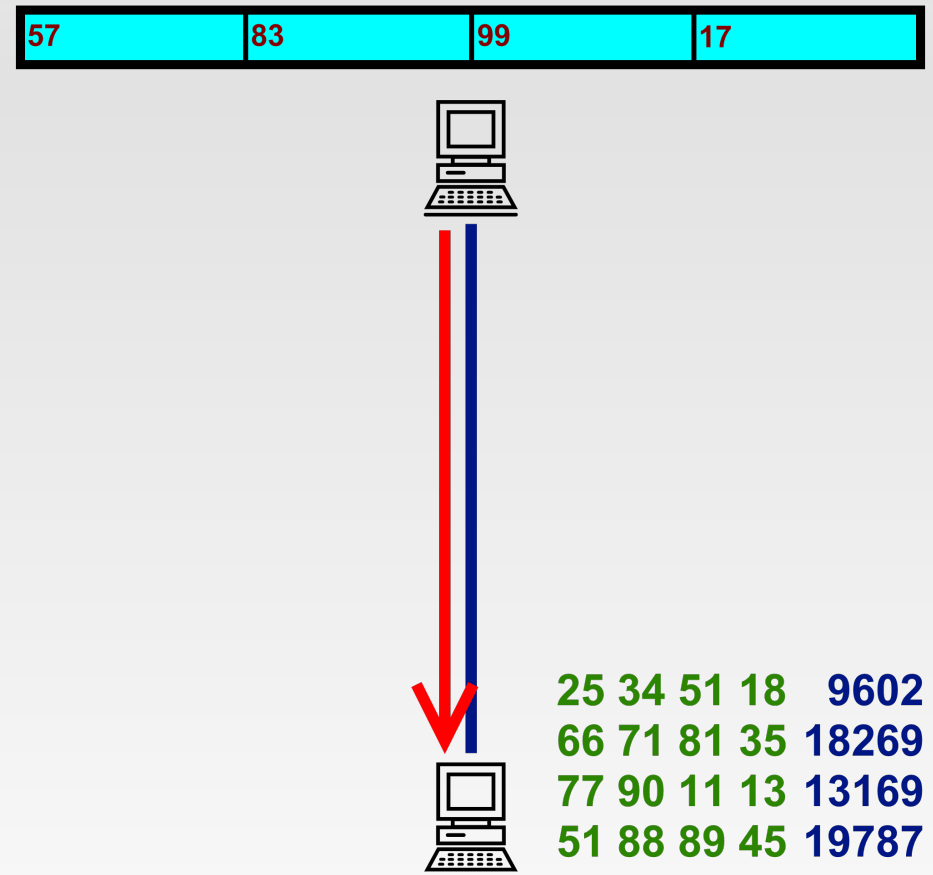- Transmits a linear combination of all of the blocks in the file, along with these coefficients.

- After receiving *n* of these, a peer can decode the original file.

Network Coding

- Transmits a linear combination of all of the blocks in the file, along with these coefficients.
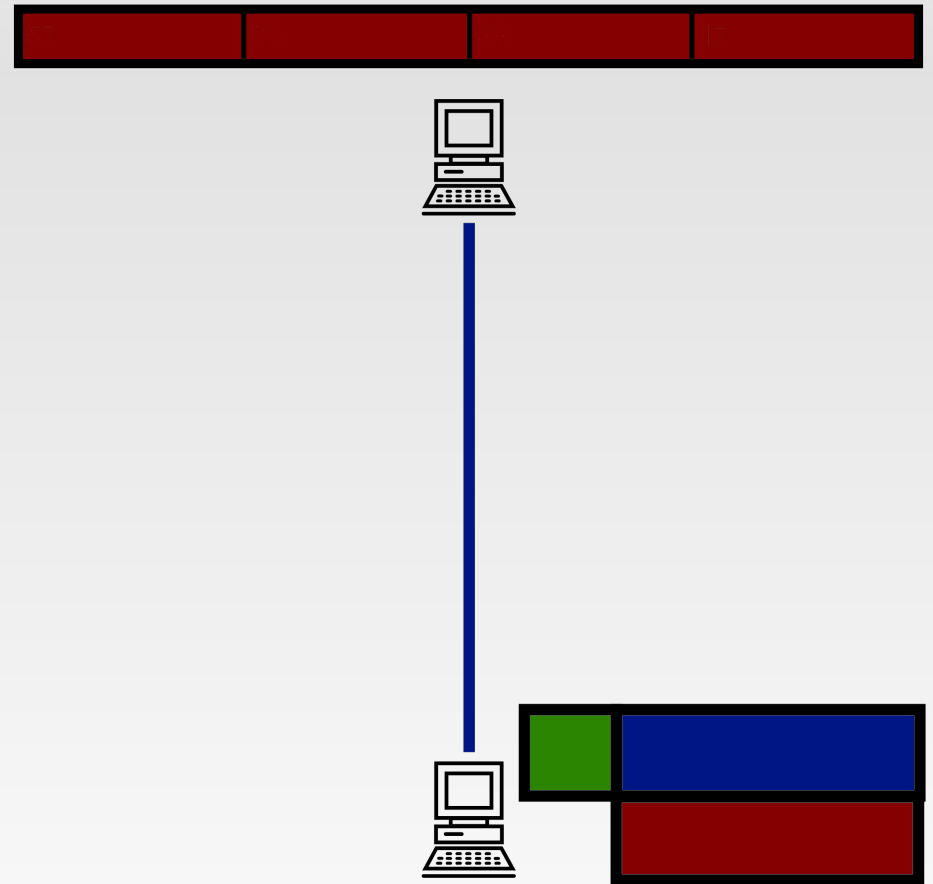
- After receiving *n* of these, a peer can decode the original file.
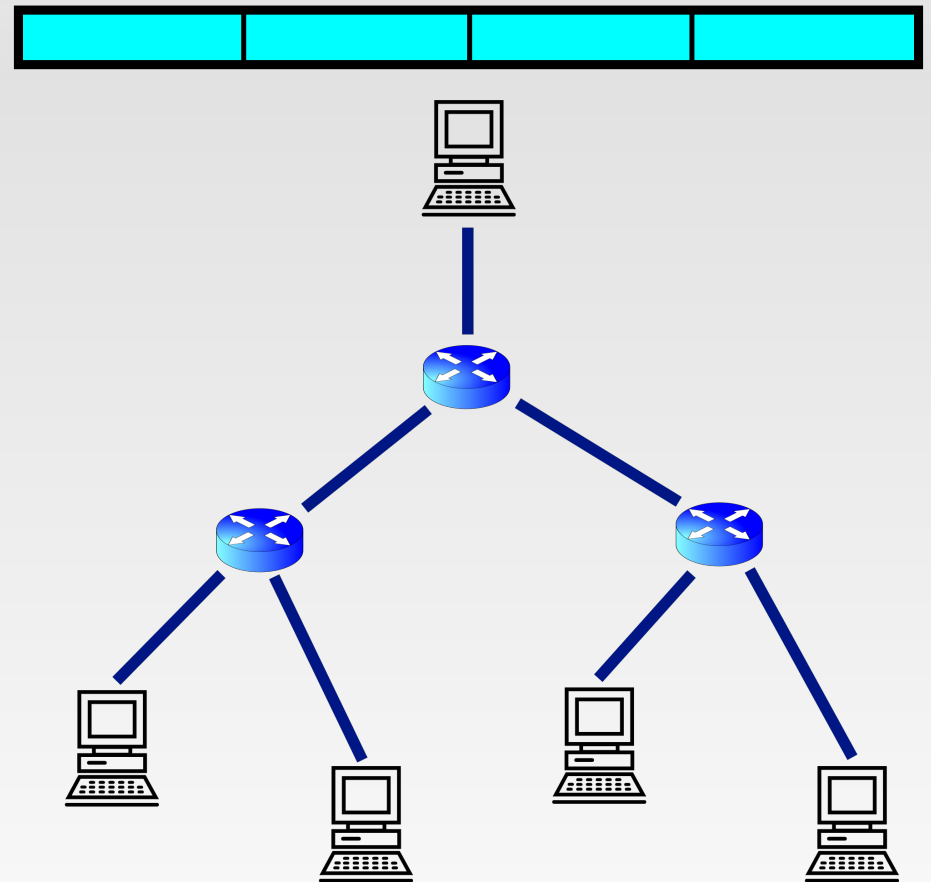
Finite Fields

- Closed over regular arithmetic operations (addition, subtraction, multiplication, division)

- All operations reversible

- Finite number of elements

Finite Fields - F_2

```
+ | 0 1
--+----
0 | 0 1
1 | 1 0
```

```
× | 0 1
--+----
0 | 0 0
1 | 0 1
```

Finite Fields - F_3

```
+ | 0 1 2          × | 0 1 2
--+------          --+------
0 | 0 1 2          0 | 0 0 0
1 | 1 2 0          1 | 0 1 2
2 | 2 0 1          2 | 0 2 1
```

Galois Field

- Meets our finite field criteria.

- Addition and subtraction operations are identical.

- Multiplication and division can be performed with table lookups.

- GF_8 and GF_{16} are common choices for network coding implementations.

BitTorrent vs. Network Coding

- Peers no longer concerned with who has which blocks.

- Reduces wasted network resources in certain situations.

- Higher computational overhead.

- Difficult to verify received blocks.

BitTorrent vs. Network Coding
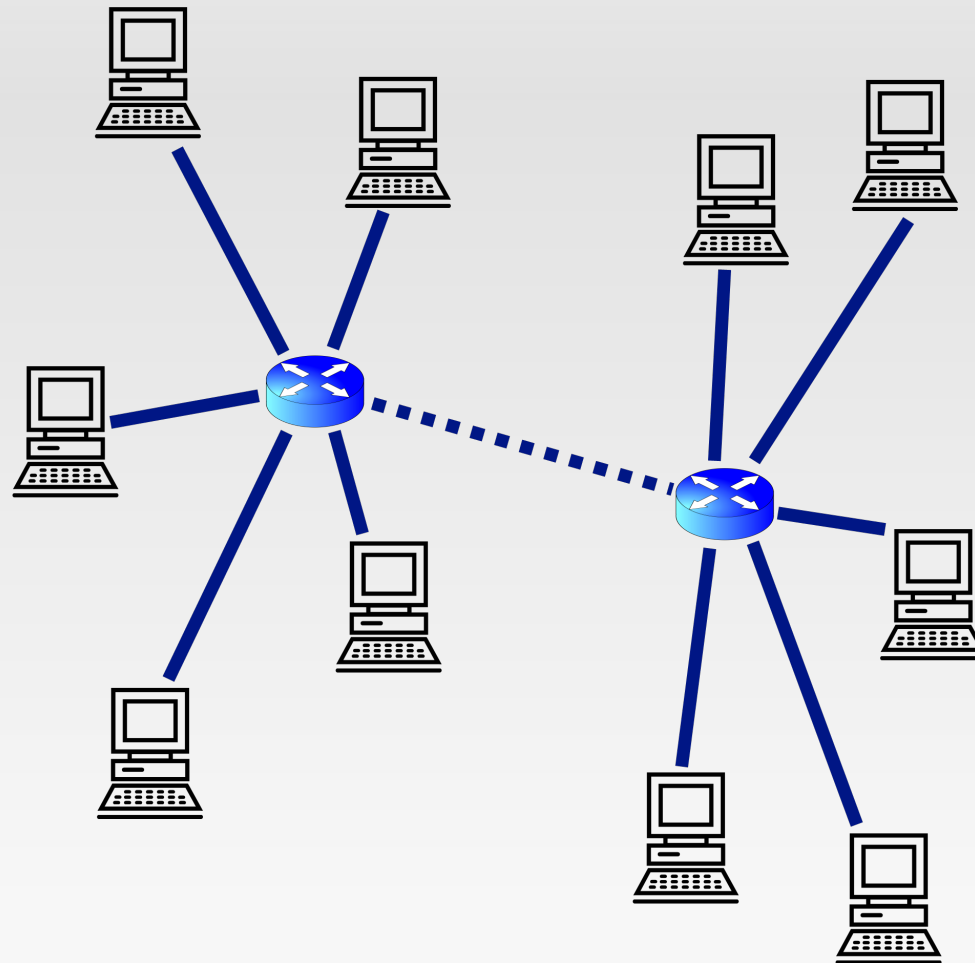
Making Network Coding More Practical?

- If we can reduce the computational overhead of network coding, it becomes more practical for widespread use.

GF$_8$ Matrix Multiplication

- Multiplying two *n* x *n* matrices – how many operations are required?

  - Resulting matrix contains n^2 elements.

  - Each of these requires *n* multiplications and additions to compute.

  - n^3 multiplications, additions.

18	38	19	72	25		41	25	36	87	99
47	46	28	91	54		66	32	11	54	76
71	56	35	84	41		33	16	50	75	19
15	23	66	98	55		34	73	65	42	88
62	28	42	31	90		91	22	54	76	31

?	?	?	?	?
?	?	?	?	?
?	?	?	?	?
?	?	?	?	?
?	?	?	?	?

GF₈ Matrix Multiplication

- We're in a finite field – can we use this to lower the number of operations?

18	38	19	72	25		41	25	36	87	99
47	46	28	91	54		66	32	11	54	76
71	56	35	84	41		33	16	50	75	19
15	23	66	98	55		34	73	65	42	88
62	28	42	31	90		91	22	54	76	31

| ?	?	?	?	?
?	?	?	?	?
?	?	?	?	?
?	?	?	?	?
?	?	?	?	?

*GF*₈ Matrix Multiplication

- Precompute all possible multiples of each column of *A*.
  - 2^F x *n* for each column
- All multiplications are done!
- Addition in *GF* is the XOR operation.
  - Multiple values can be added simultaneously.

18	38	19	72	25	41	25	36	87	99
47	46	28	91	54	66	32	11	54	76
71	56	35	84	41	33	16	50	75	19
15	23	66	98	55	34	73	65	42	88
62	28	42	31	90	91	22	54	76	31

?	?	?	?	?
?	?	?	?	?
?	?	?	?	?
?	?	?	?	?
?	?	?	?	?

How does this help network coding?

- Decode
  - Invert coefficient matrix, multiply by data matrix to recover original data.

- Encode
  - Generate an entire matrix of random values, multiply by file to get *n* blocks to send.

- Both of these can be implemented as a series of square matrix multiplications.
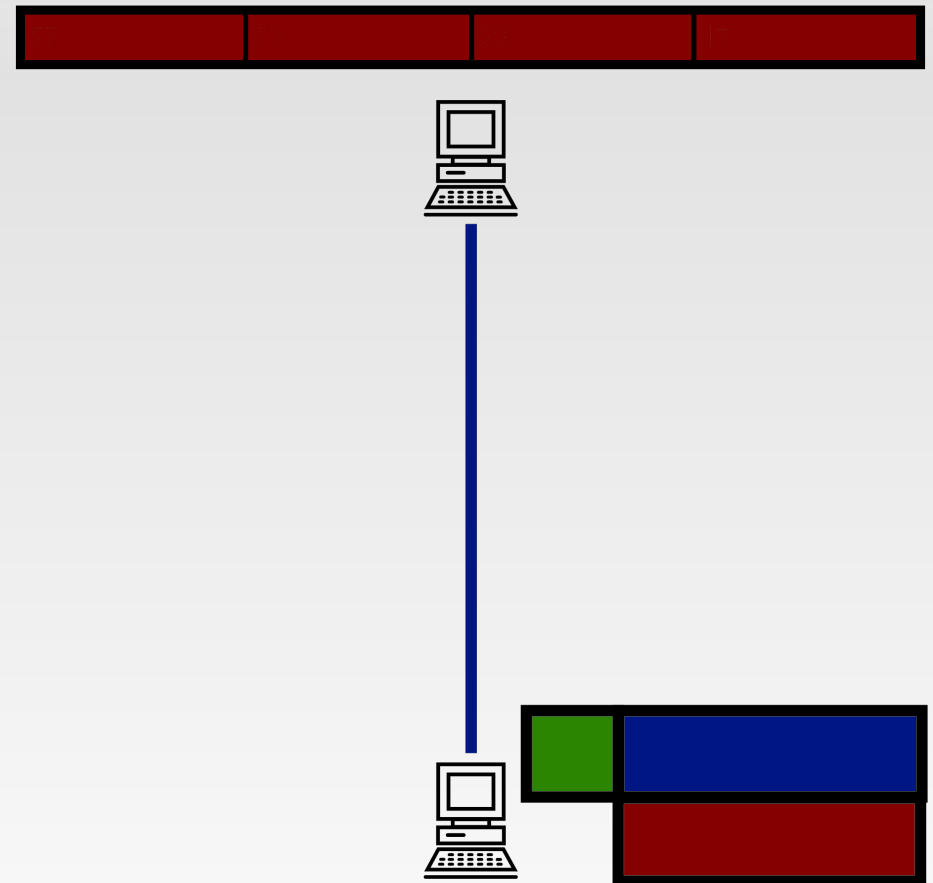
Preliminary Results

- How much faster is this matrix multiplication?
  - 2048 x 2048 matrix
  - GF_8
  - Naïve algorithm:
    - 186 seconds
  - Precomputation:
    - 12.5 seconds precomputation
    - 3.5 seconds of addition

Questions

?