On the Connection Between Multiple-Unicast Network Coding and Single-Source Single-Sink Network Error Correction

Jörg Kliewer
NJIT

Joint work with Wentao Huang and Michael Langberg

Network Error Correction

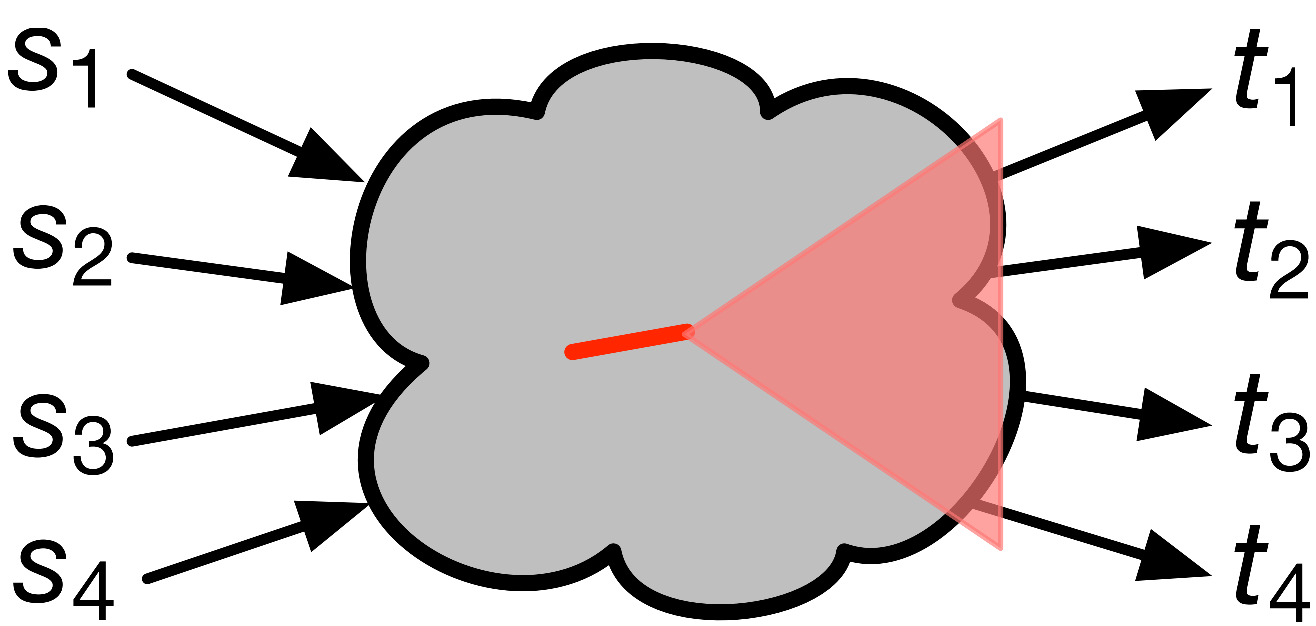
Problem:

- Adversary has control of some edges in a network
- Objective: Design coding scheme that is resilient against adversary
- Which rates are achievable?

Network Error Correction

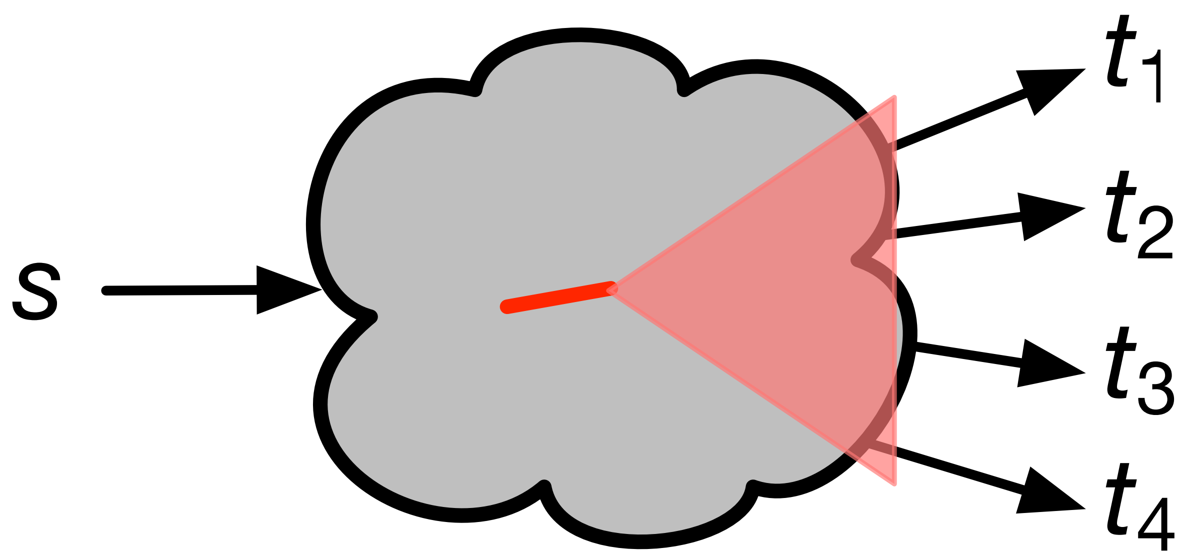
Problem:

- Adversary has control of some edges in a network
- Objective: Design coding scheme that is resilient against adversary
- Which rates are achievable?



Known Cases

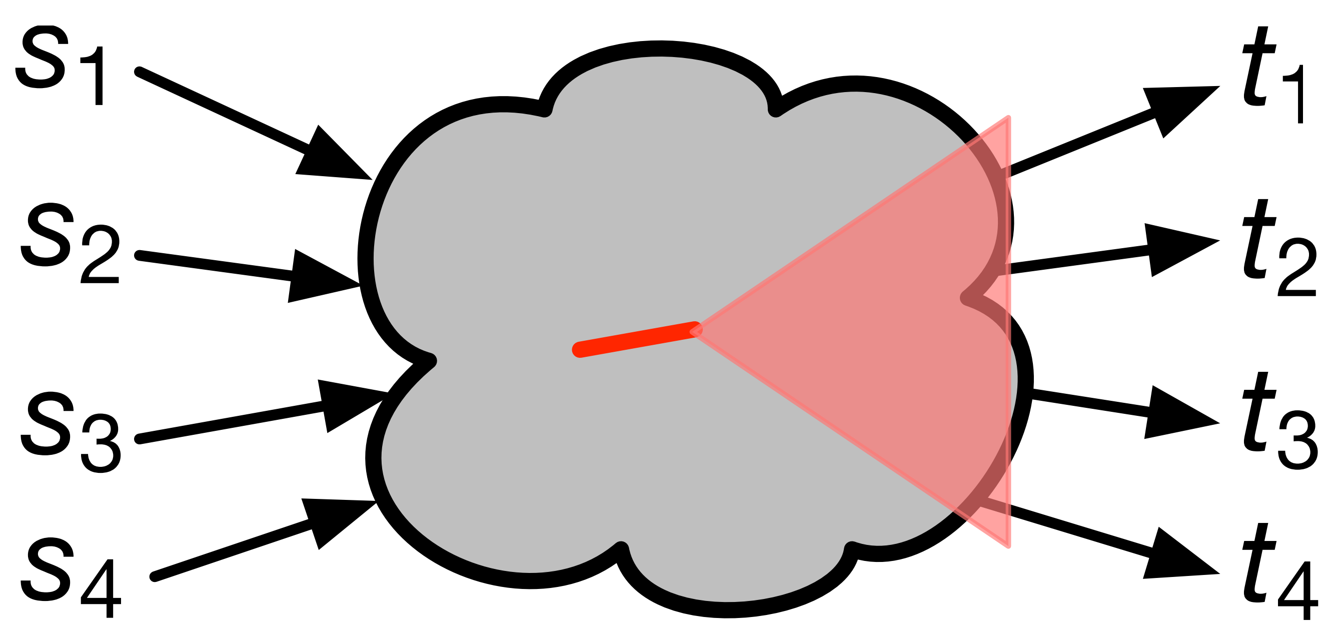
- Adversary controls z links in the network
- **Single source multicast**, equal capacity links
  - Capacity: mincut – $2z$
  - Code design, e.g., in [Cai & Yeung 06], [Koetter & Kschischang 08], [Jaggi et al. 08], [Silva et al. 08], [Brito, Kliewer 13]



Less Known and Studied Cases

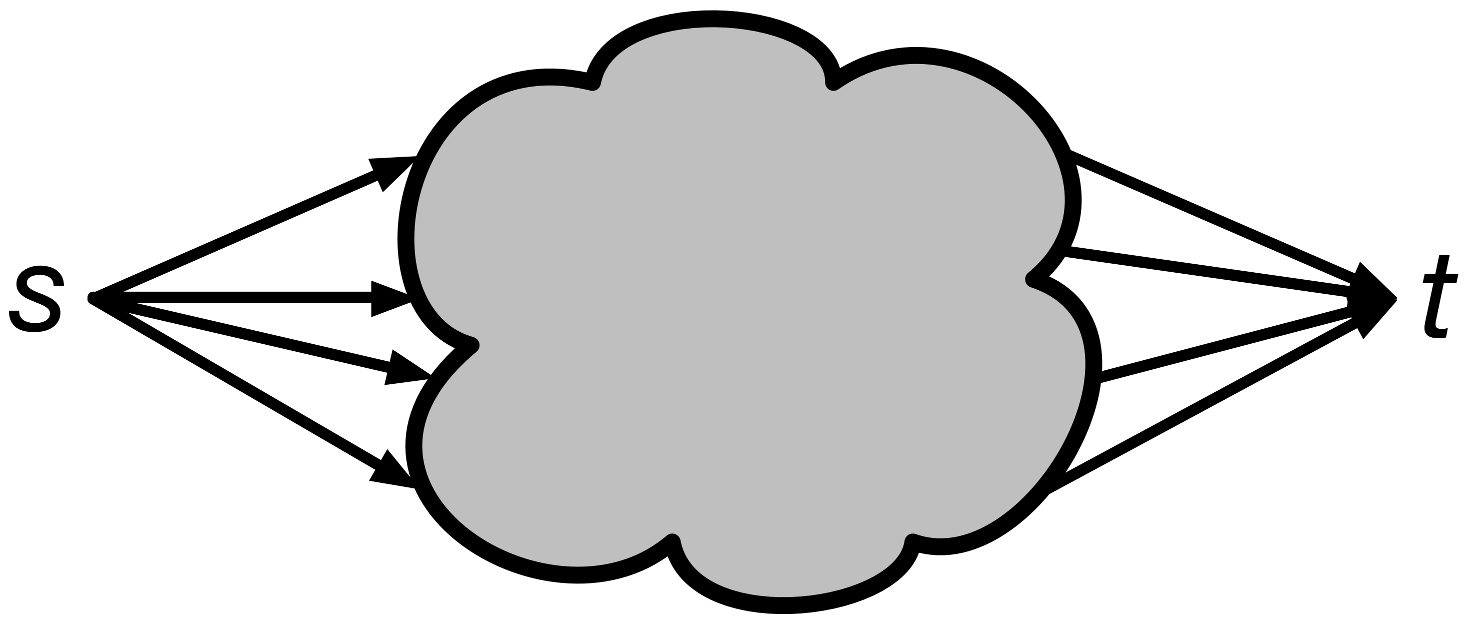
- **Single source multicast:** different edge capacities, node adversaries, restricted adversaries (e.g., [Kosut, Tong, Tse 09], [Kim et al. 11], [Wang, Silva, Kschischang 08])

- **Multiple sources and terminals:** Upper and lower capacity bounds [Vyetrenko, Ho, Dikaliotis 10], [Liang, Agrawal, Vaidya 10]



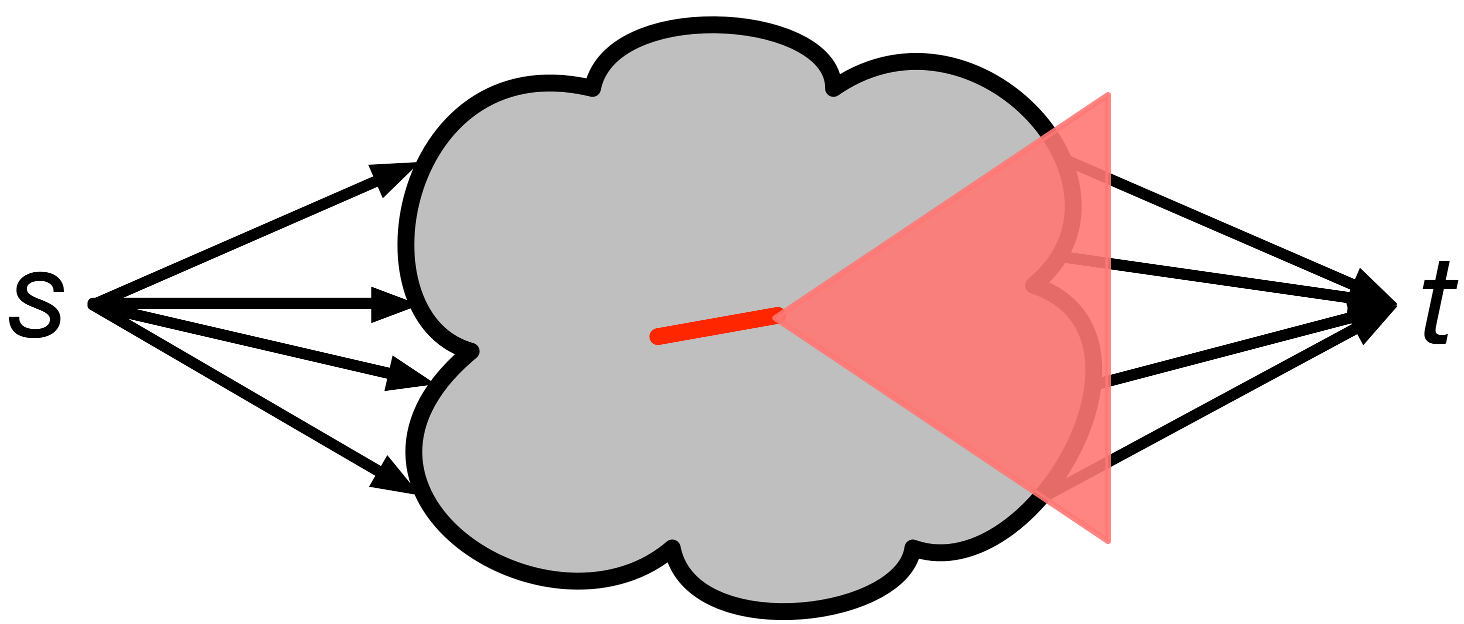
This Work

- Single-source single-sink
- Acyclic network
- Edges may not have unit capacity



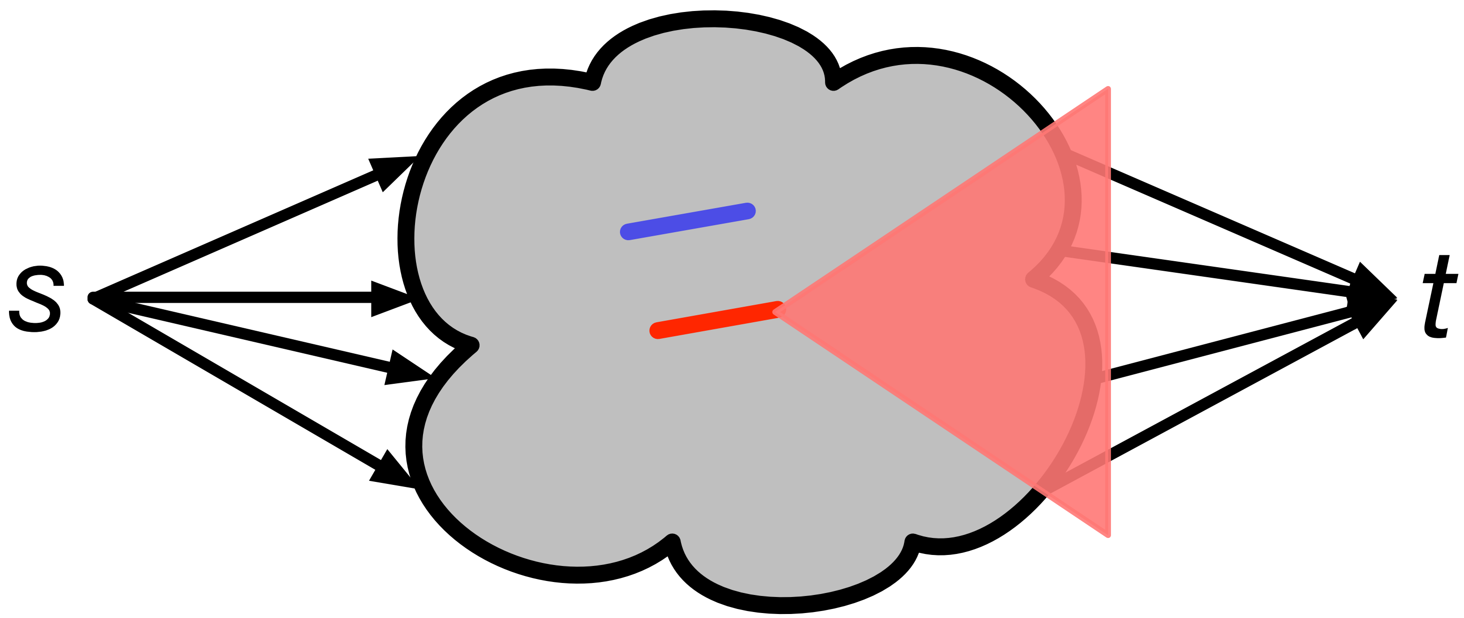
This Work

- Single-source single-sink
- Acyclic network
- Edges may not have unit capacity
- Adversary controls single link



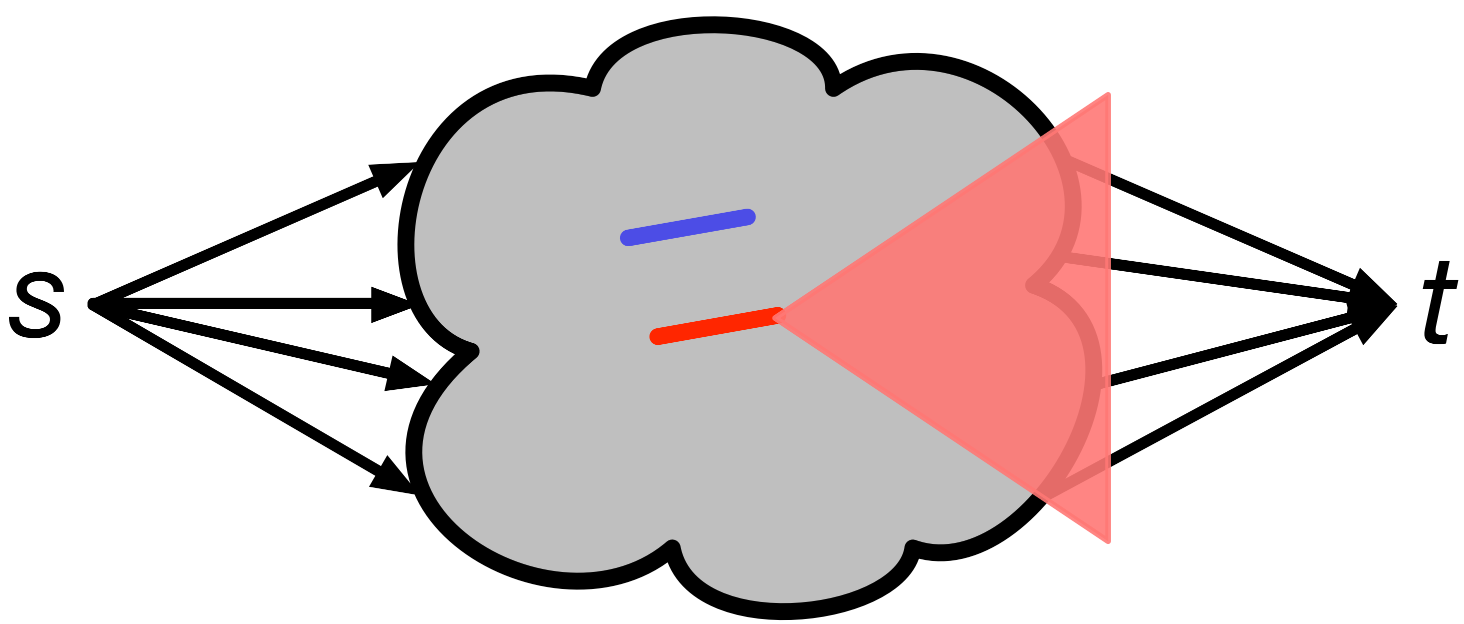
This Work

- Single-source single-sink
- Acyclic network
- Edges may not have unit capacity
- Adversary controls single link
- Some edges cannot be accessed by the adversary



This Work

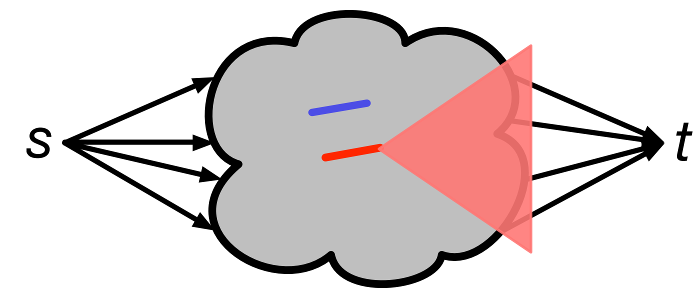
- Single-source single-sink
- Acyclic network
- Edges may not have unit capacity
- Adversary controls single link
- Some edges cannot be accessed by the adversary
- **Reliable communication rate?**



Results

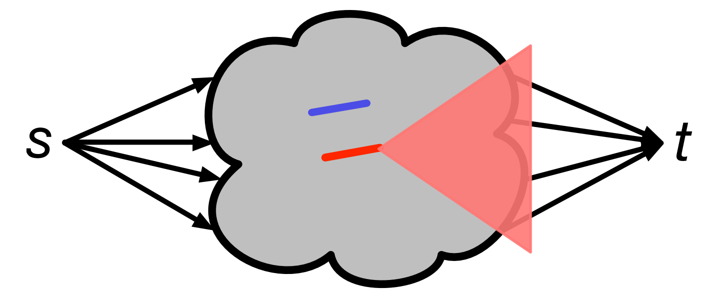
Results

Network error correction problem:



Results

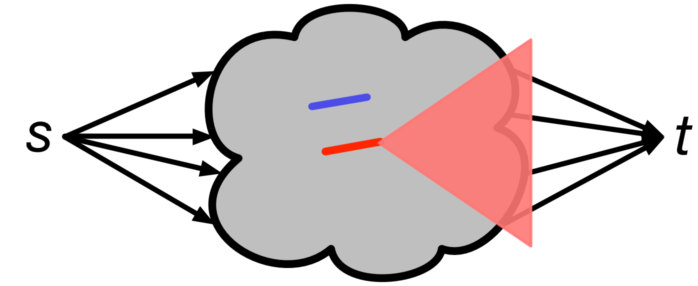
Network error correction problem:



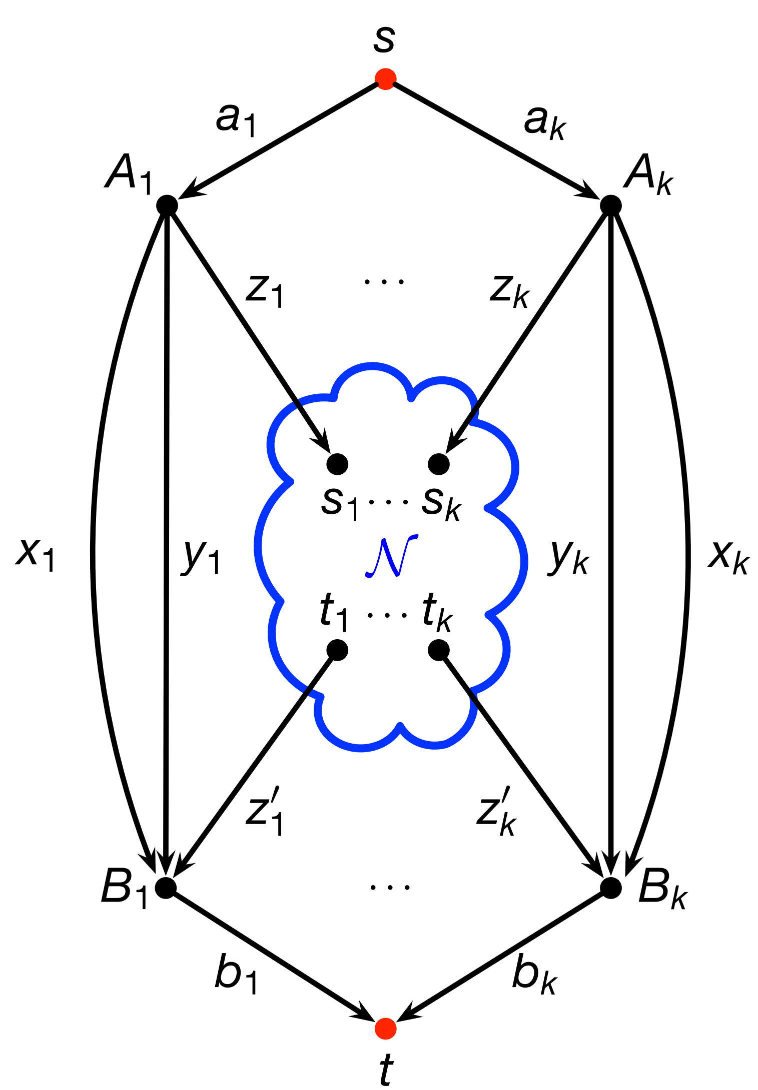
Computing capacity is as hard as computing the capacity of an error-free multiple unicast network coding problem.

Results

Network error correction problem:

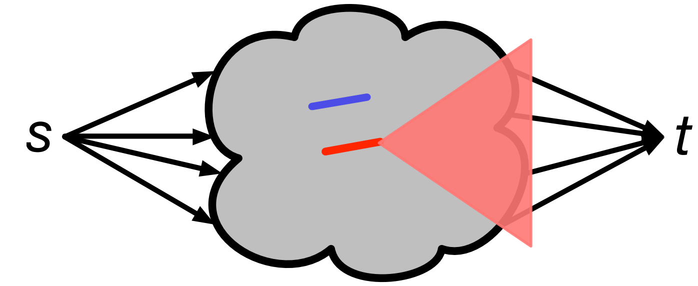


Computing capacity is as hard as computing the capacity of an error-free multiple unicast network coding problem.

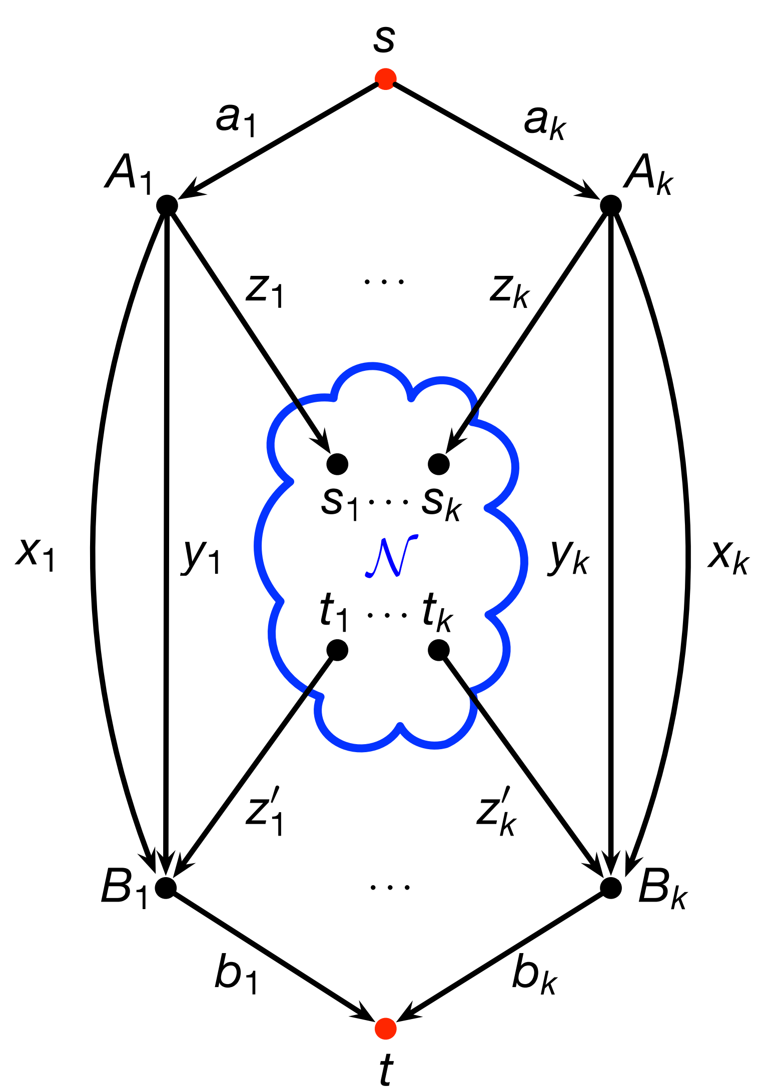


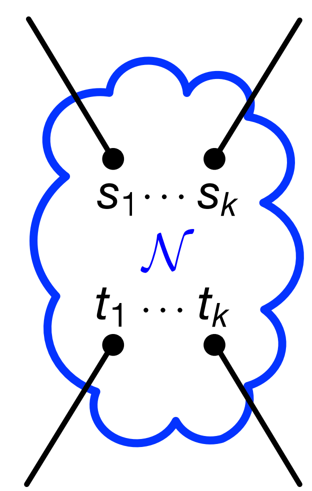
Results

Network error correction problem:

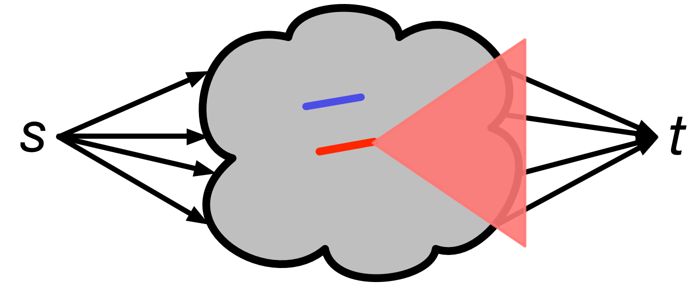


Computing capacity is as hard as computing the capacity of an error-free multiple unicast network coding problem.

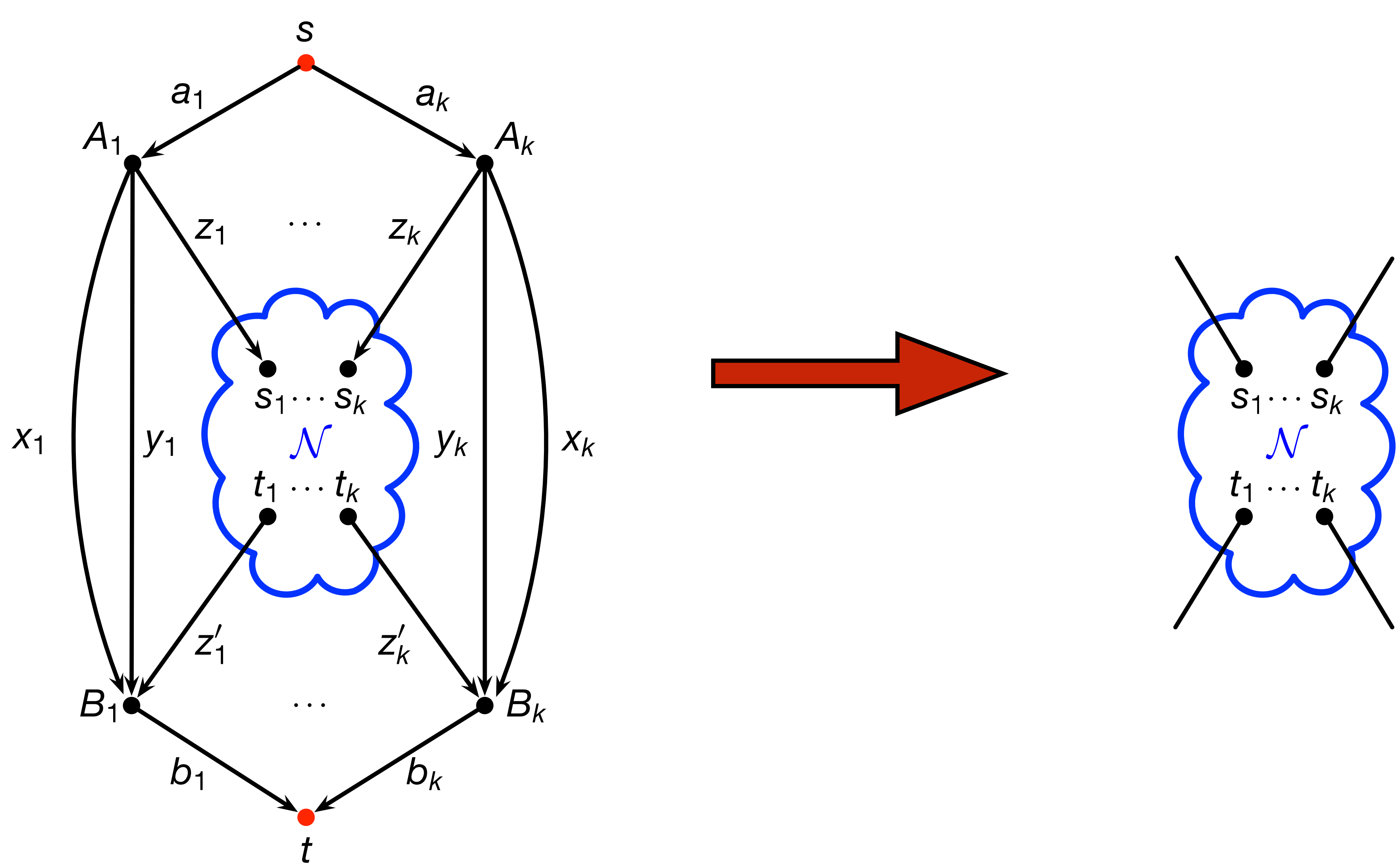




Results

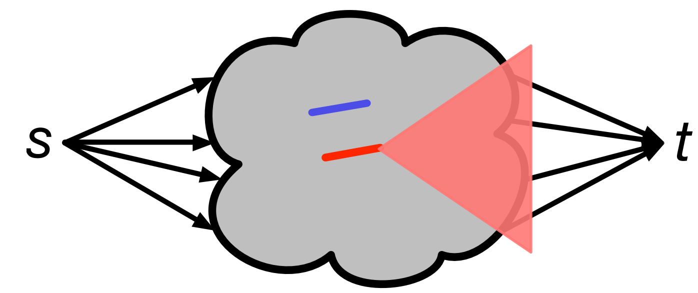
Network error correction problem: 

Computing capacity is as hard as computing the capacity of an error-free multiple unicast network coding problem.

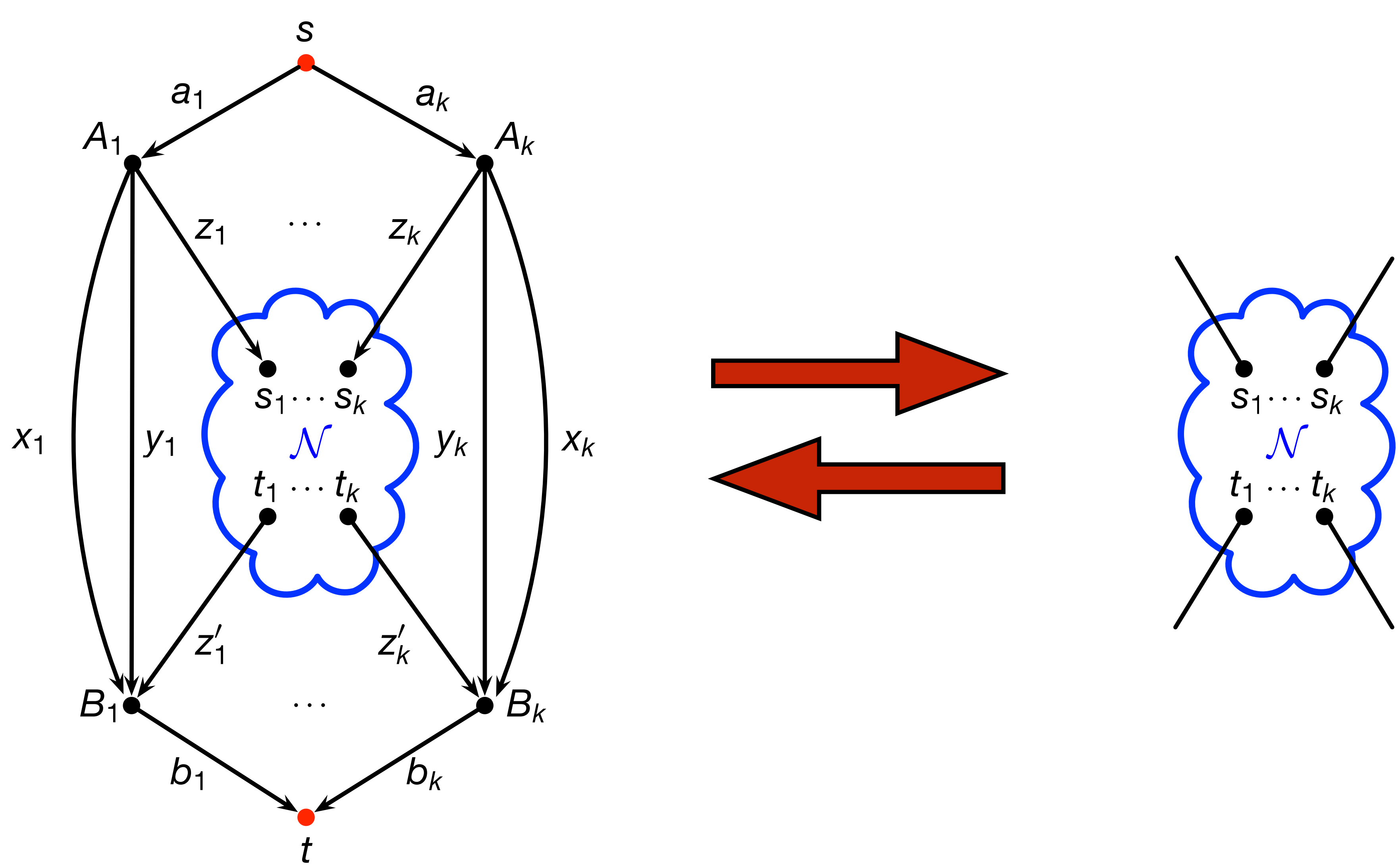


Results

Network error correction problem:



Computing capacity is as hard as computing the capacity of an error-free multiple unicast network coding problem.

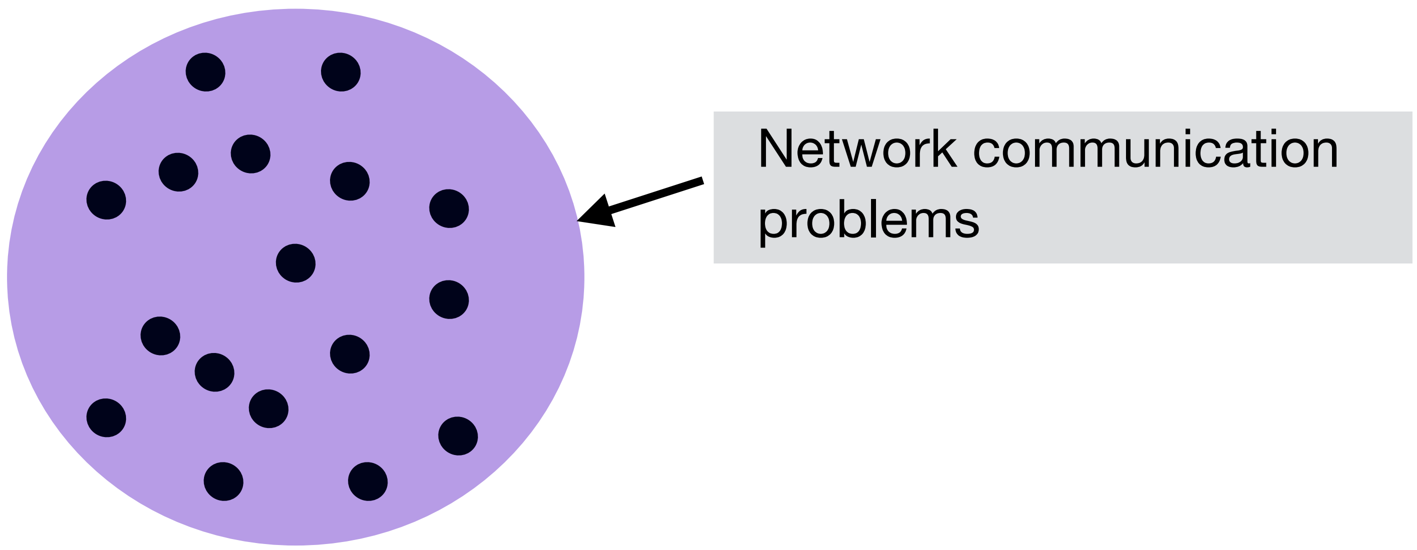


Reductions

- Result proved by reduction
- Significant interest recently
  - Index coding/network coding, index coding/interference alignment, network equivalence, multiple unicast vs. multiple multicast NC, edge removal problem, …

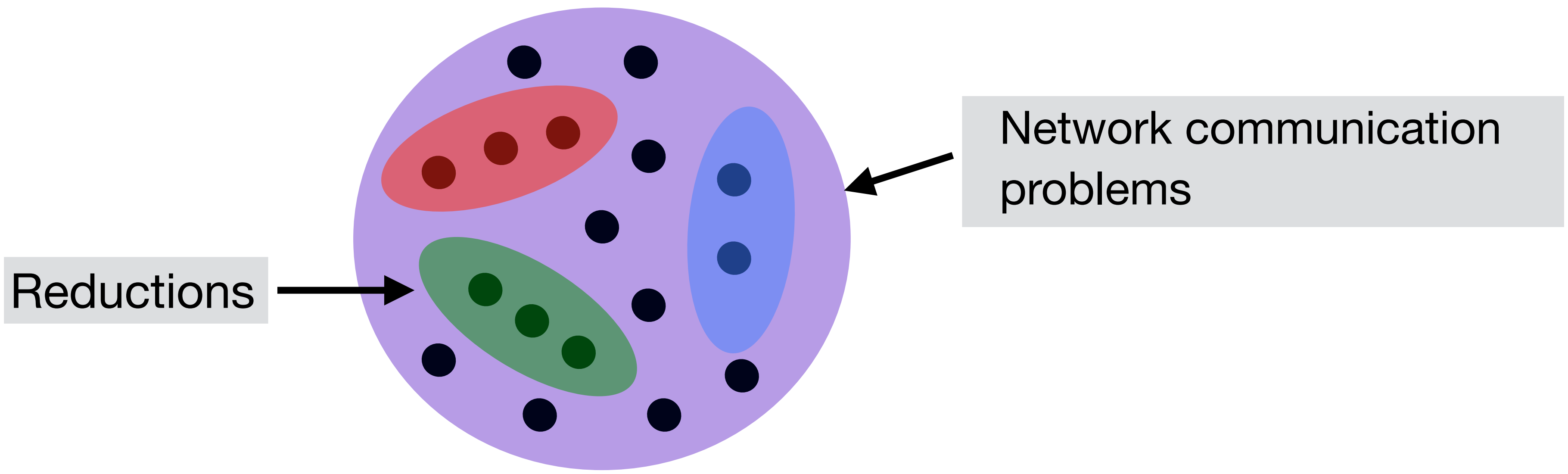
Reductions

- Result proved by reduction
- Significant interest recently
  - Index coding/network coding, index coding/interference alignment, network equivalence, multiple unicast vs. multiple multicast NC, edge removal problem, …
- Generate a clustering of network communication problems via reductions



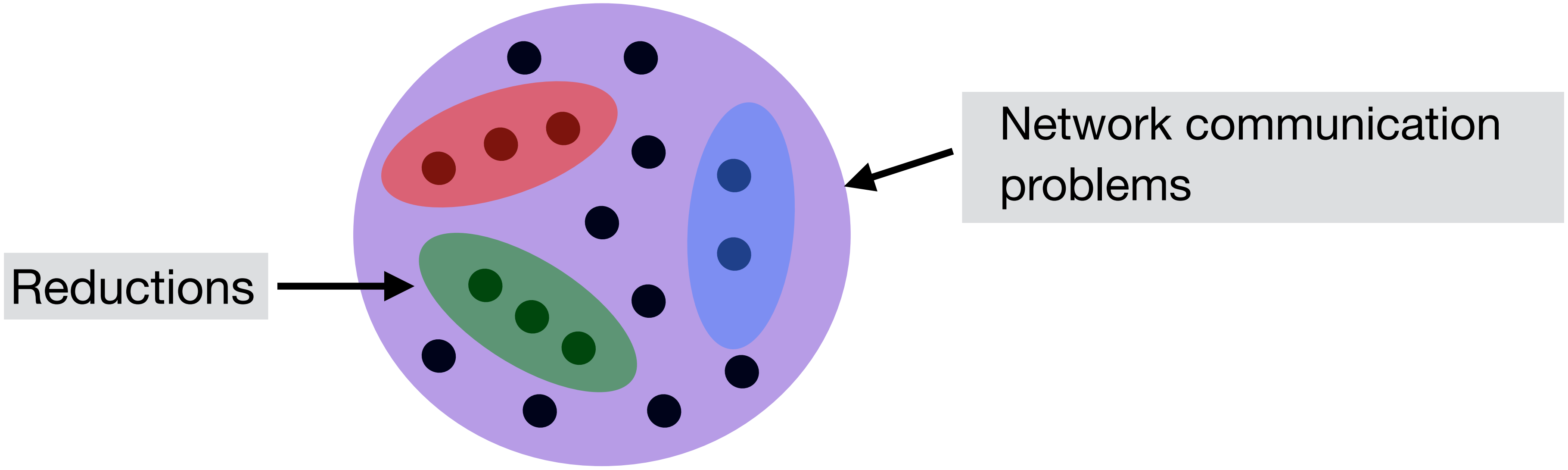
Reductions

- Result proved by reduction
- Significant interest recently
  - Index coding/network coding, index coding/interference alignment, network equivalence, multiple unicast vs. multiple multicast NC, edge removal problem, …
- Generate a clustering of network communication problems via reductions



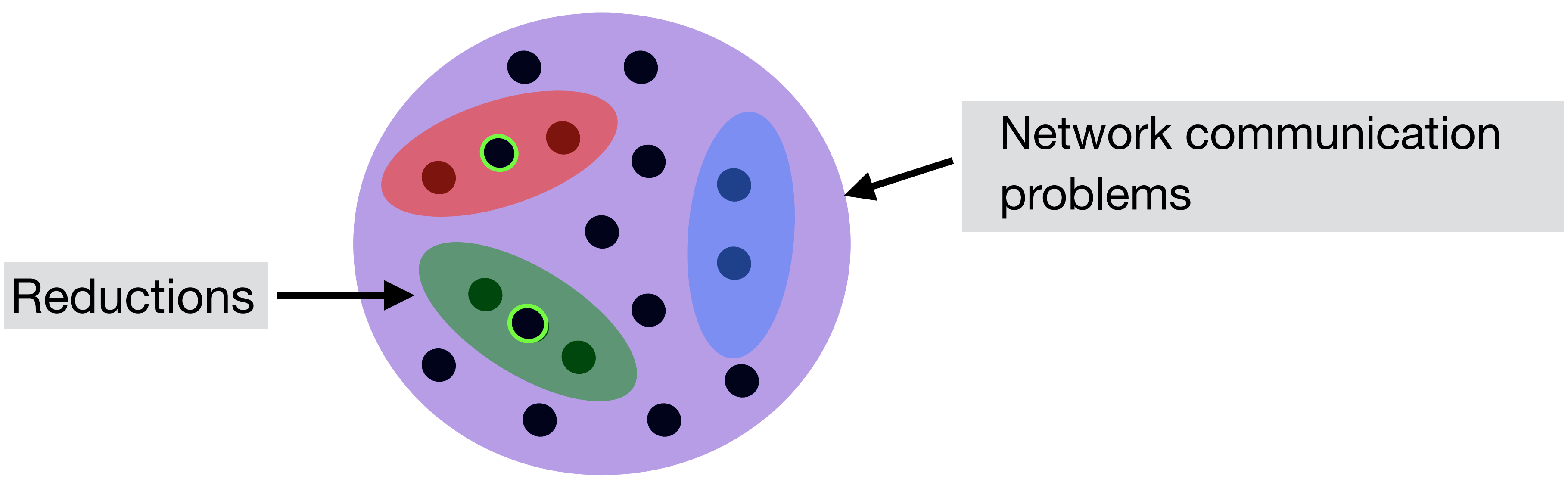
Reductions

- Result proved by reduction
- Significant interest recently
    - Index coding/network coding, index coding/interference alignment, network equivalence, multiple unicast vs. multiple multicast NC, edge removal problem, …
- Generate a clustering of network communication problems via reductions
- Identify canonical problems (central problems that are related to several other problems)



Reductions

- Result proved by reduction
- Significant interest recently
  - Index coding/network coding, index coding/interference alignment, network equivalence, multiple unicast vs. multiple multicast NC, edge removal problem, …
- Generate a clustering of network communication problems via reductions
- Identify canonical problems (central problems that are related to several other problems)



Proof

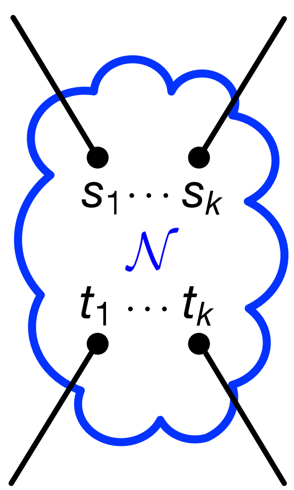
Computing error correcting capacity is as hard as computing the capacity of an error-free multiple unicast network coding problem.

- Proof for zero error communication
- Result also holds for both the case of asymptotic rate and asymptotic error

Proof

Computing error correcting capacity is as hard as computing the capacity of an error-free multiple unicast network coding problem.

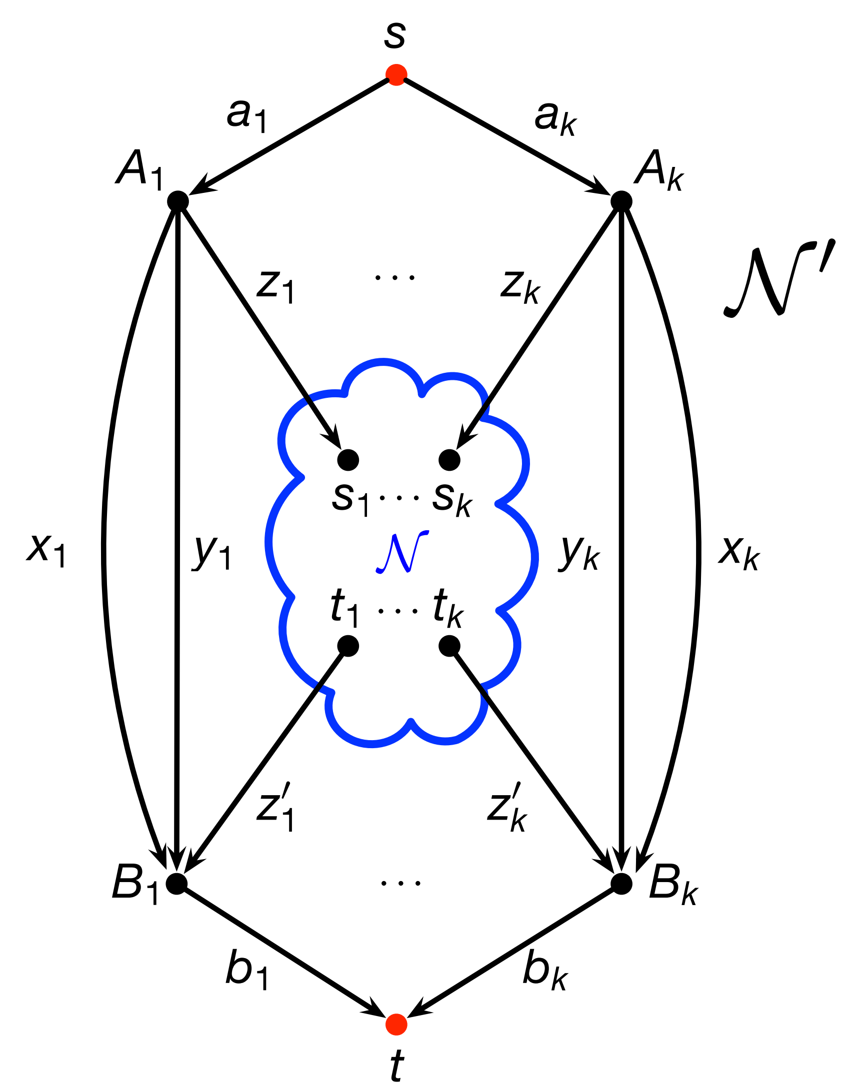
- Proof for zero error communication
- Result also holds for both the case of asymptotic rate and asymptotic error
- Starting point: MU NC problem $\mathcal{N}$
- Is rate tuple $(1, 1, \ldots, 1)$ achievable with zero error?



Proof

Computing error correcting capacity is as hard as computing the capacity of an error-free multiple unicast network coding problem.

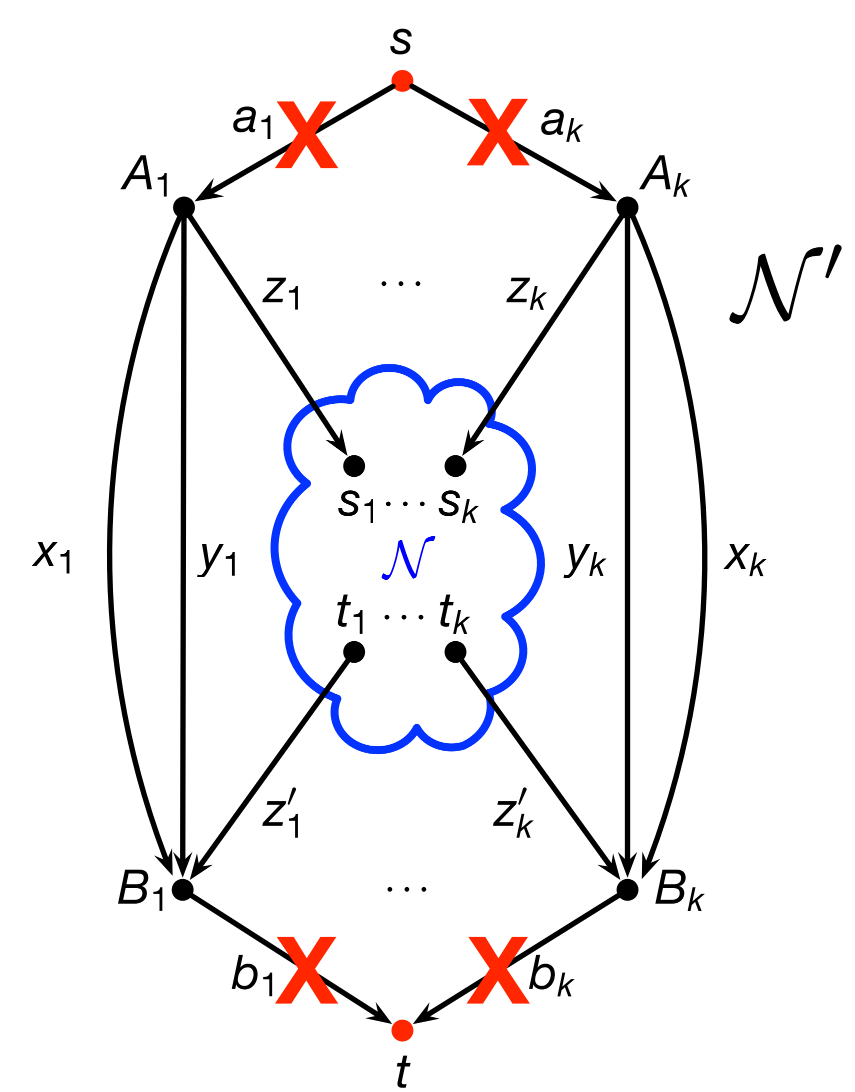
- Proof for zero error communication
- Result also holds for both the case of asymptotic rate and asymptotic error
- Starting point: MU NC problem $\mathcal{N}$
- Is rate tuple $(1, 1, \ldots, 1)$ achievable with zero error?
- Reduction: Construct new network $\mathcal{N}'$



Proof

Computing error correcting capacity is as hard as computing the capacity of an error-free multiple unicast network coding problem.

- Proof for zero error communication
- Result also holds for both the case of asymptotic rate and asymptotic error
- Starting point: MU NC problem $\mathcal{N}$
- Is rate tuple $(1, 1, \ldots, 1)$ achievable with zero error?
- Reduction: Construct new network $\mathcal{N}'$
- Adversary can access any single link except links leaving *s* and *t*



Zero Error Case

Theorem

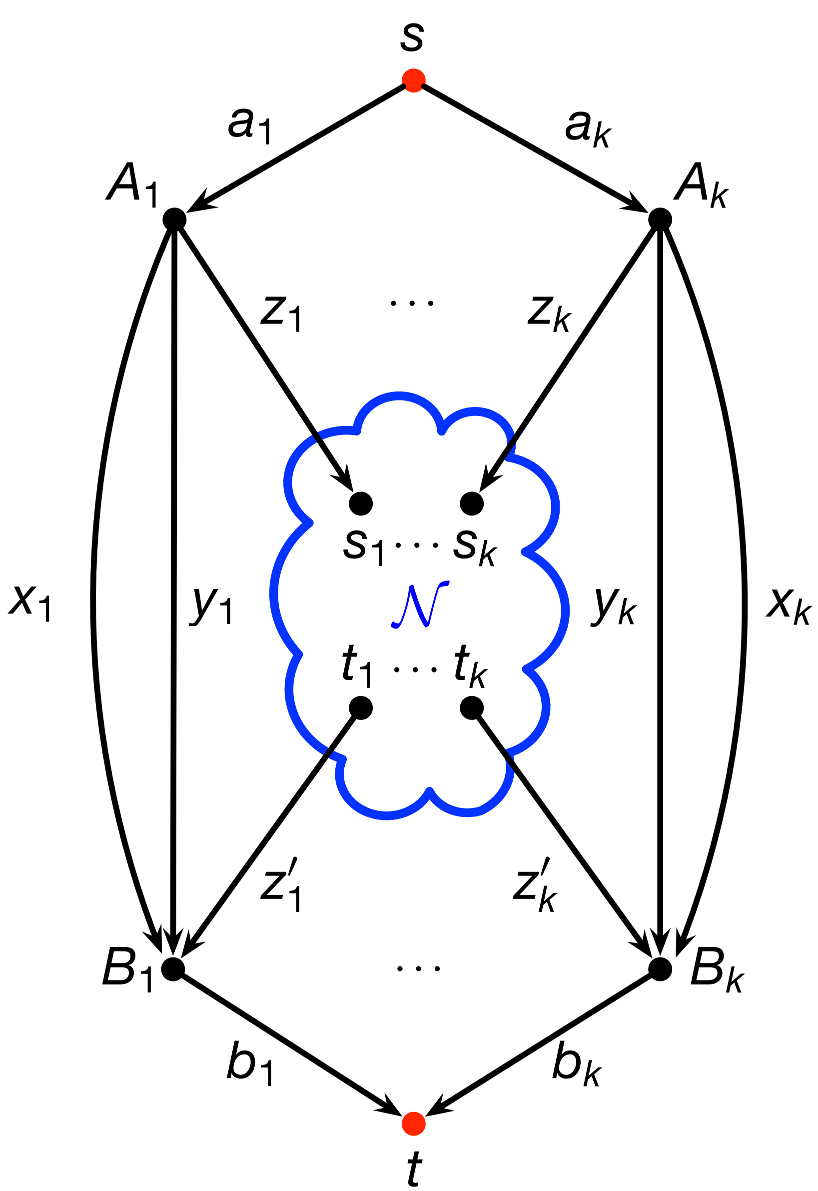
Rates $(1, 1, \ldots, 1)$ achievable on $\mathcal{N}$ iff rate *k* is achievable on $\mathcal{N}'$.

Zero Error Case

Theorem

Rates $(1, 1, \ldots, 1)$ achievable on $\mathcal{N}$ iff rate k is achievable on $\mathcal{N}'$.

Proof sketch: ⇐



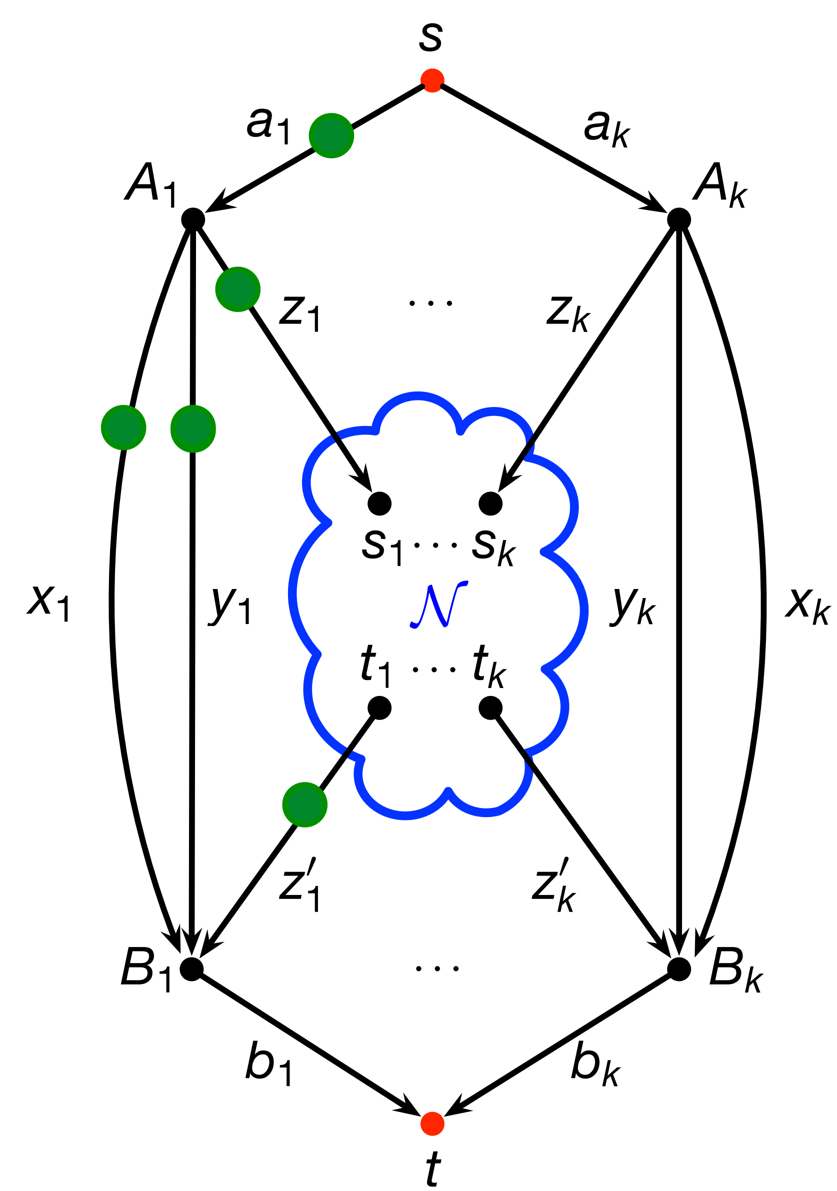
Zero Error Case

Theorem

Rates $(1, 1, \ldots, 1)$ achievable on $\mathcal{N}$ iff rate k is achievable on $\mathcal{N}'$.

Proof sketch: ⇐

- Assume $(1, 1, \ldots, 1)$ on $\mathcal{N}$
- Source sends information on a_i



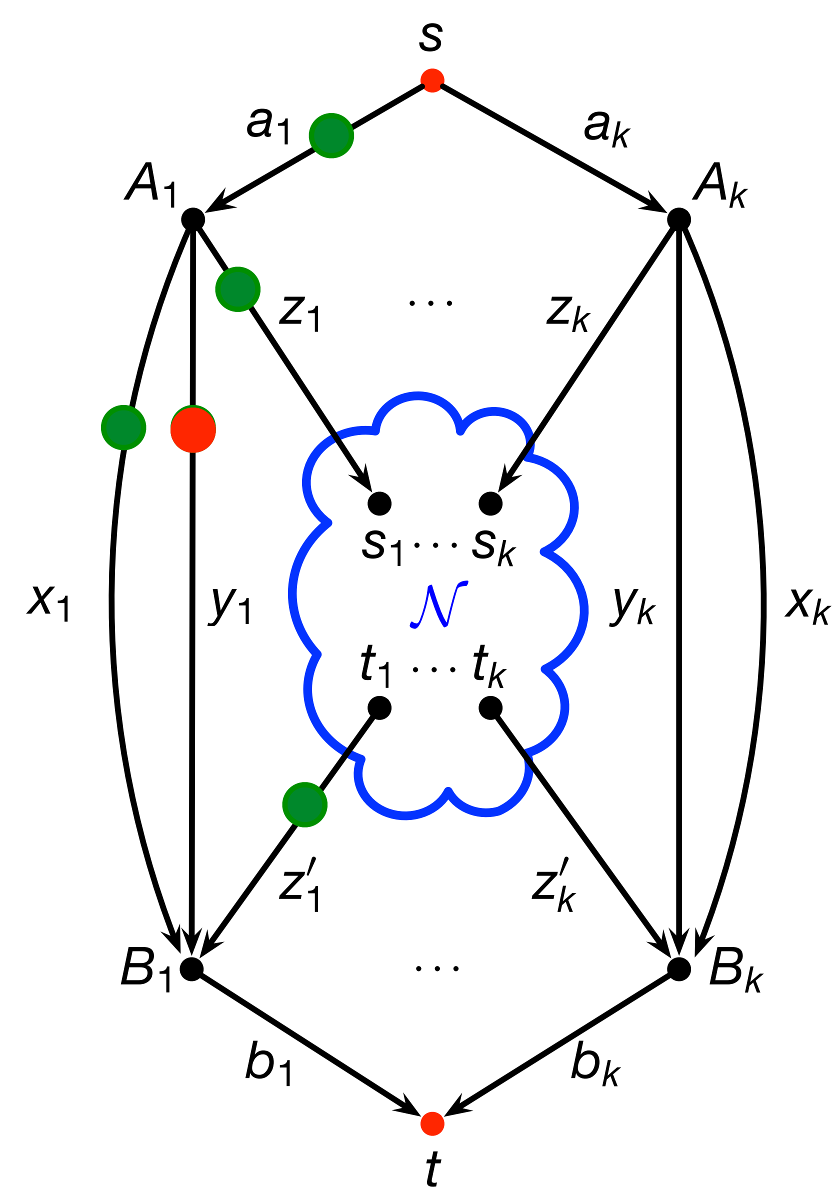
Zero Error Case

Theorem

Rates $(1, 1, \ldots, 1)$ achievable on $\mathcal{N}$ iff rate k is achievable on $\mathcal{N}'$.

Proof sketch:

- Assume $(1, 1, \ldots, 1)$ on $\mathcal{N}$
- Source sends information on a_i
- One error may occur on x_i, y_i, z_i, z_i'



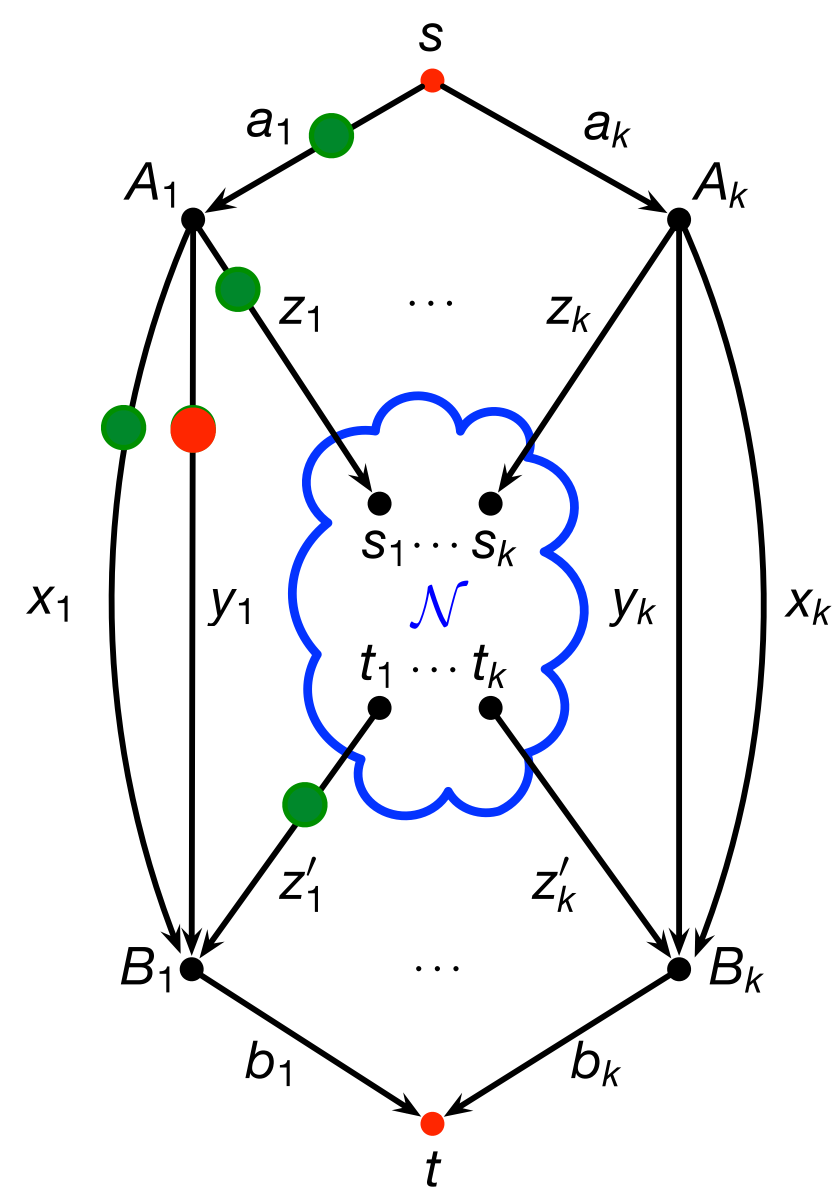
Zero Error Case

Theorem

Rates $(1, 1, \ldots, 1)$ achievable on $\mathcal{N}$ iff rate k is achievable on $\mathcal{N}'$.

Proof sketch: ⇐

- Assume $(1, 1, \ldots, 1)$ on $\mathcal{N}$
- Source sends information on a_i
- One error may occur on x_i, y_i, z_i, z_i'
- B_i performs majority decoding
- Rate k is possible on $\mathcal{N}'$



Zero Error Case

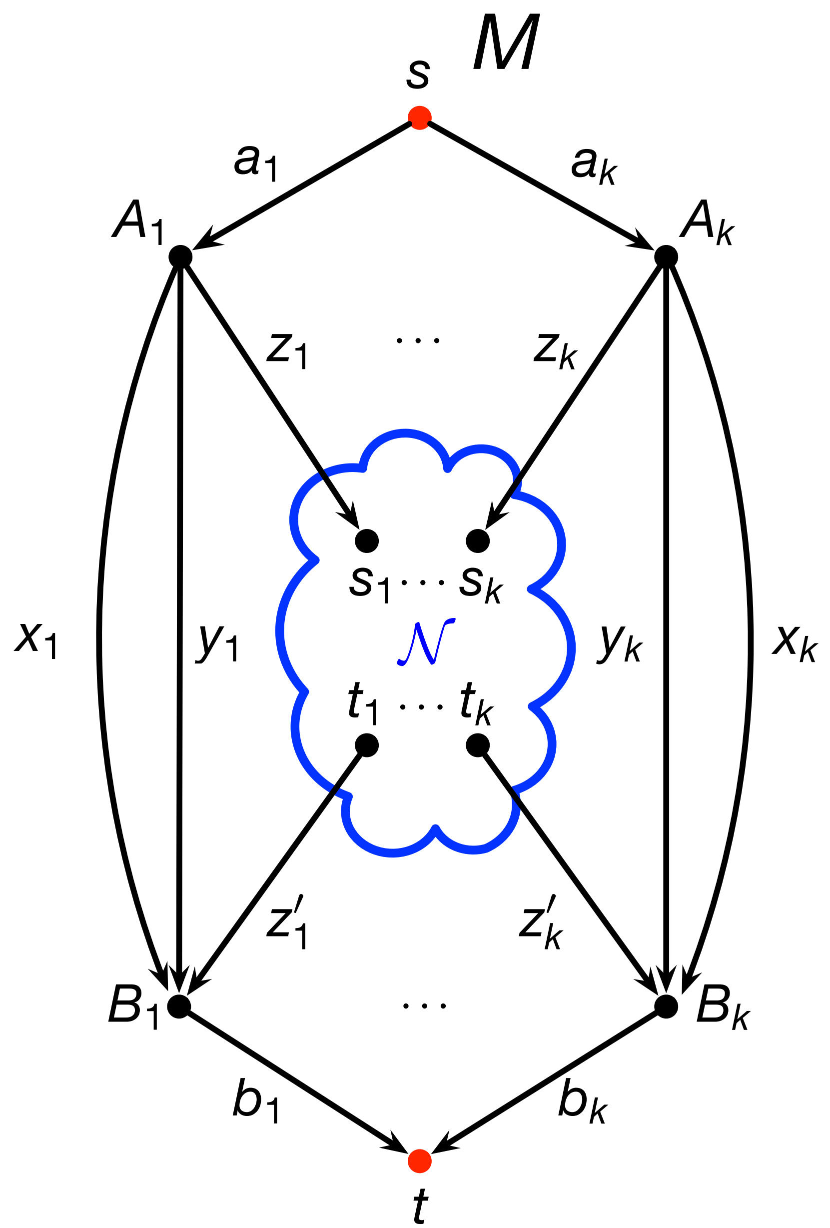
Zero Error Case

Proof sketch: ➡

Zero Error Case

Proof sketch: ➡

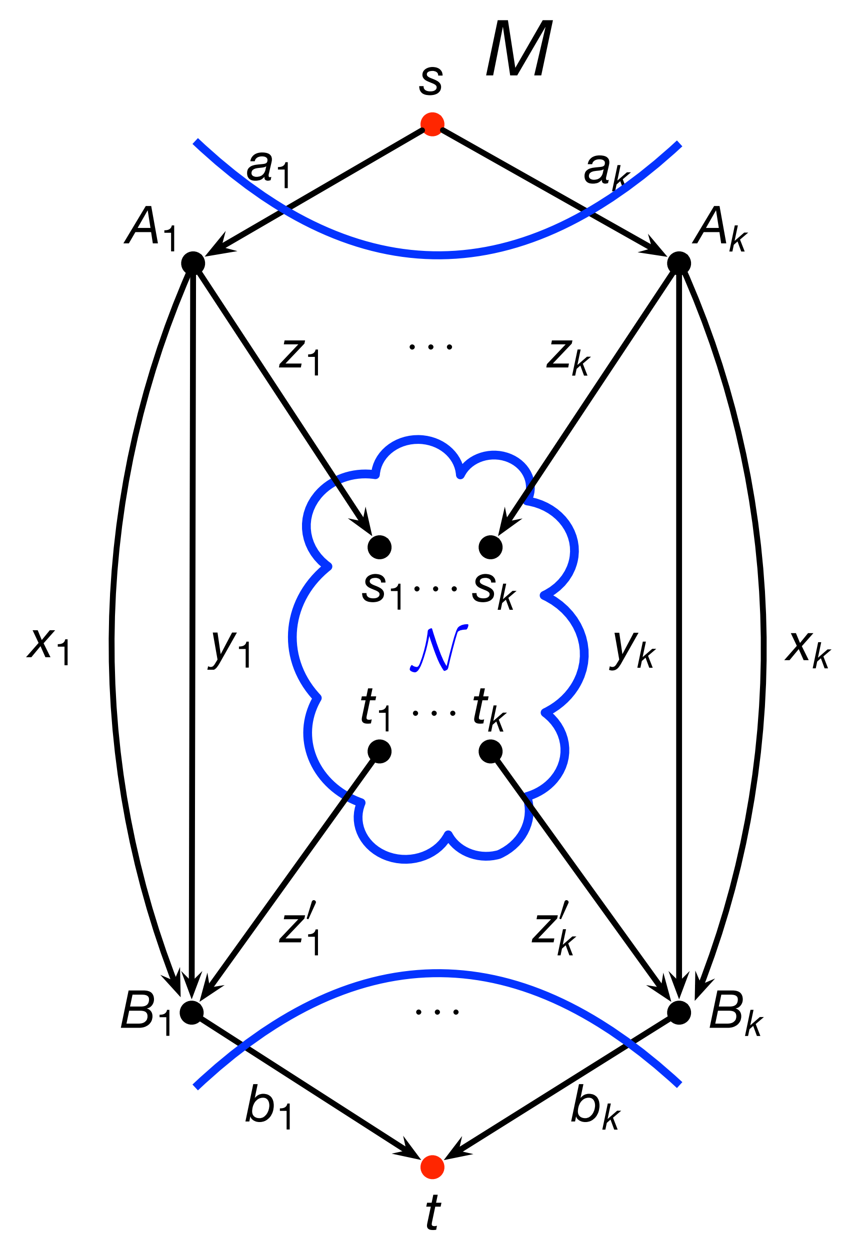
- Assume rate k achievable on $\mathcal{N}'$: show rate $(1, 1, \ldots, 1)$ on $\mathcal{N}$



Zero Error Case

Proof sketch: ➡

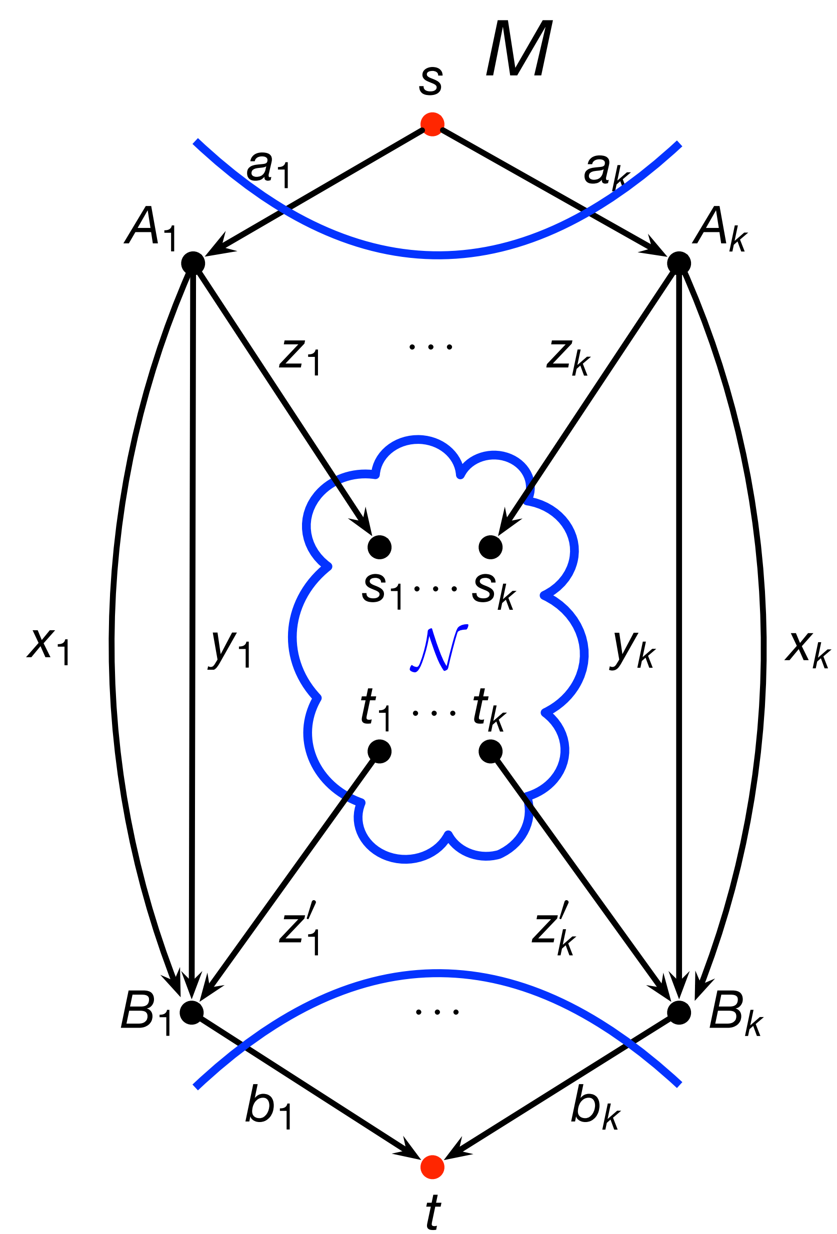
- Assume rate k achievable on $\mathcal{N}'$: show rate $(1, 1, \ldots, 1)$ on $\mathcal{N}$
- Full rate (cut): 1-1 corresp. between message M, $a_1, \ldots, a_k$, $b_1, \ldots, b_k$



Zero Error Case

Proof sketch: ⇒

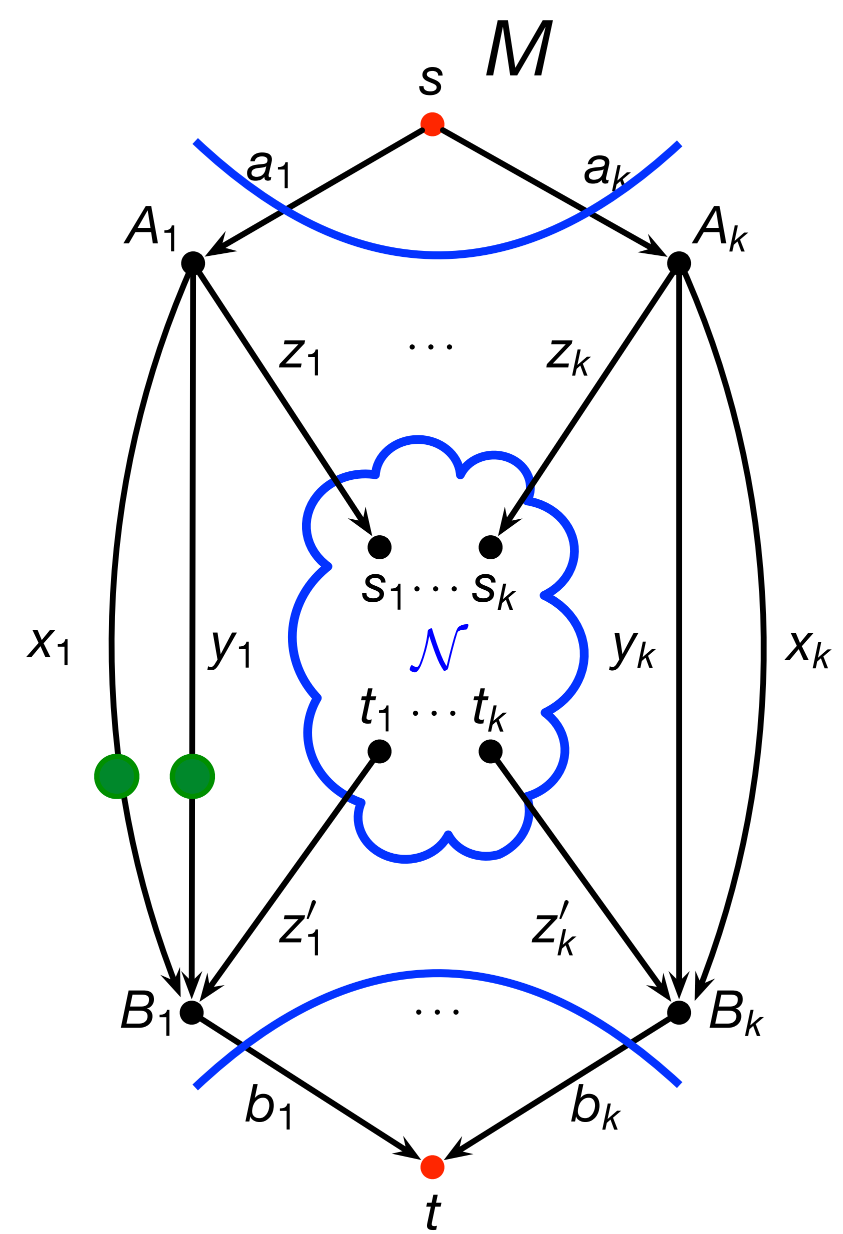
- Assume rate k achievable on $\mathcal{N}'$: show rate $(1, 1, \ldots, 1)$ on $\mathcal{N}$
- Full rate (cut): 1-1 corresp. between message M, $a_1, \ldots, a_k$, $b_1, \ldots, b_k$
- Error correction: For $M_1 \neq M_2$ if $b_i(M_1) \neq b_i(M_2)$ then $z_i'(M_1) \neq z_i'(M_2)$
- For $z_i'(M_1) = z_i'(M_2)$ terminal cannot distinguish between M_1, M_2



Zero Error Case

Proof sketch: ⇒

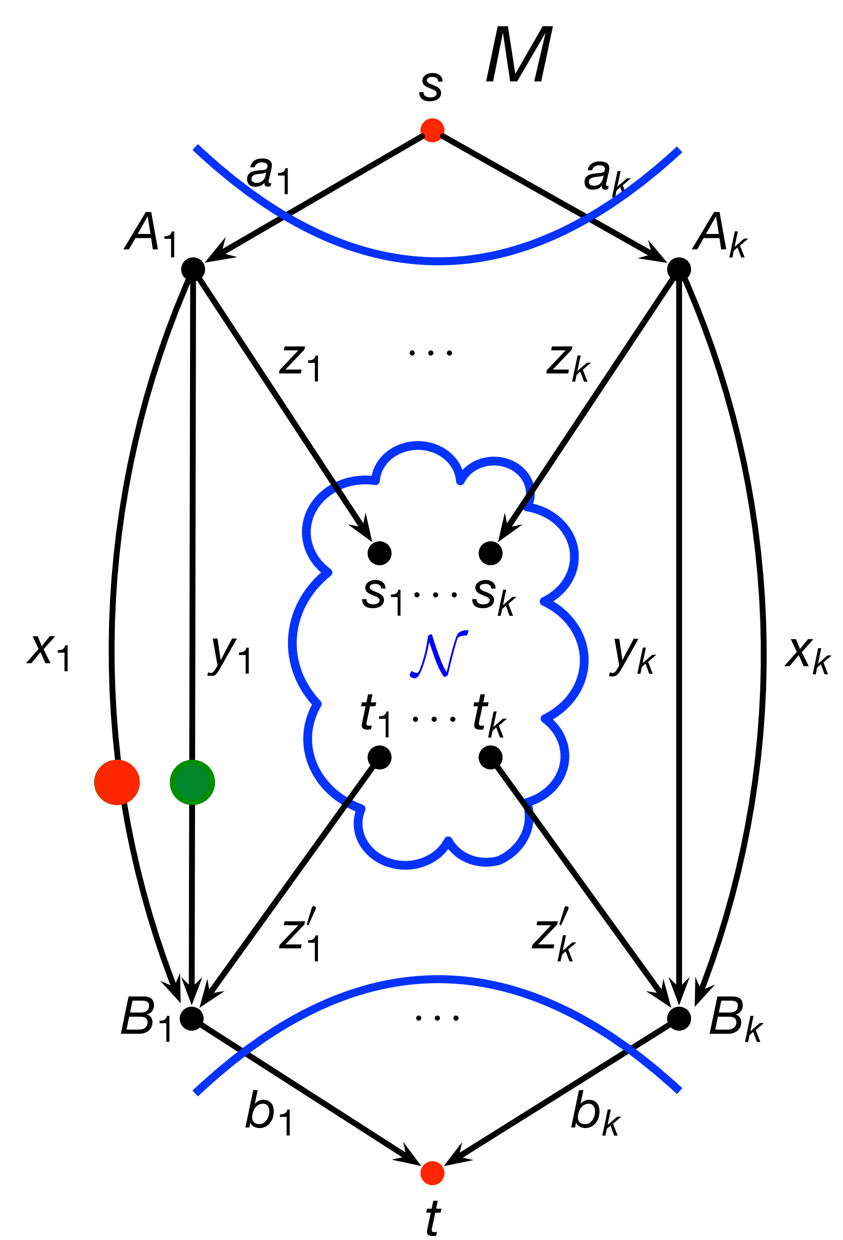
- Assume rate k achievable on $\mathcal{N}'$: show rate $(1, 1, \ldots, 1)$ on $\mathcal{N}$
- Full rate (cut): 1-1 corresp. between message M, $a_1, \ldots, a_k$, $b_1, \ldots, b_k$
- Error correction: For $M_1 \neq M_2$ if $b_i(M_1) \neq b_i(M_2)$ then $z_i'(M_1) \neq z_i'(M_2)$
- For $z_i'(M_1) = z_i'(M_2)$ terminal cannot distinguish between M_1, M_2



Zero Error Case

Proof sketch: ➡

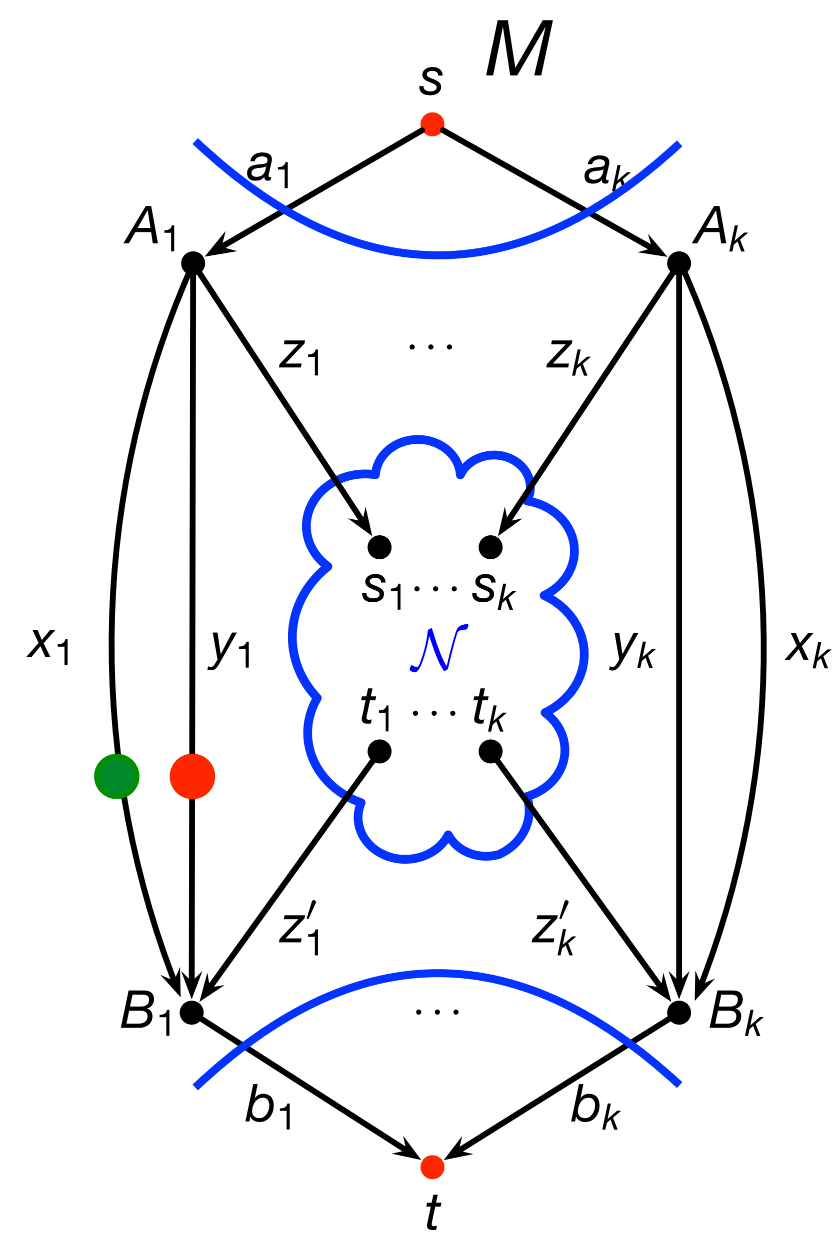
- Assume rate k achievable on $\mathcal{N}'$: show rate $(1, 1, \ldots, 1)$ on $\mathcal{N}$
- Full rate (cut): 1-1 corresp. between message M, $a_1, \ldots, a_k$, $b_1, \ldots, b_k$
- Error correction: For $M_1 \neq M_2$ if $b_i(M_1) \neq b_i(M_2)$ then $z_i'(M_1) \neq z_i'(M_2)$
- For $z_i'(M_1) = z_i'(M_2)$ terminal cannot distinguish between M_1, M_2



Zero Error Case

Proof sketch: ➡

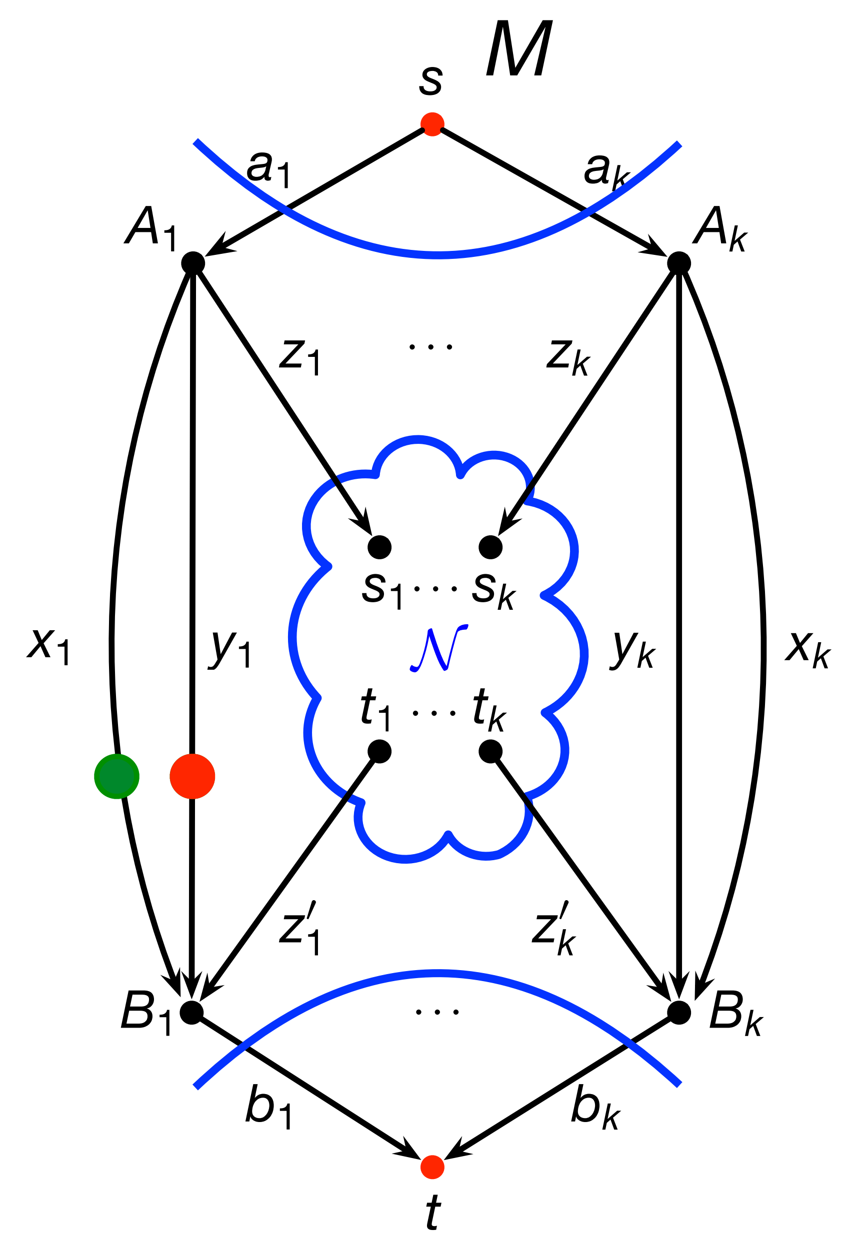
- Assume rate k achievable on $\mathcal{N}'$: show rate $(1, 1, \ldots, 1)$ on $\mathcal{N}$
- Full rate (cut): 1-1 corresp. between message M, $a_1, \ldots, a_k$, $b_1, \ldots, b_k$
- Error correction: For $M_1 \neq M_2$ if $b_i(M_1) \neq b_i(M_2)$ then $z_i'(M_1) \neq z_i'(M_2)$
- For $z_i'(M_1) = z_i'(M_2)$ terminal cannot distinguish between M_1, M_2



Zero Error Case

Proof sketch: ➡

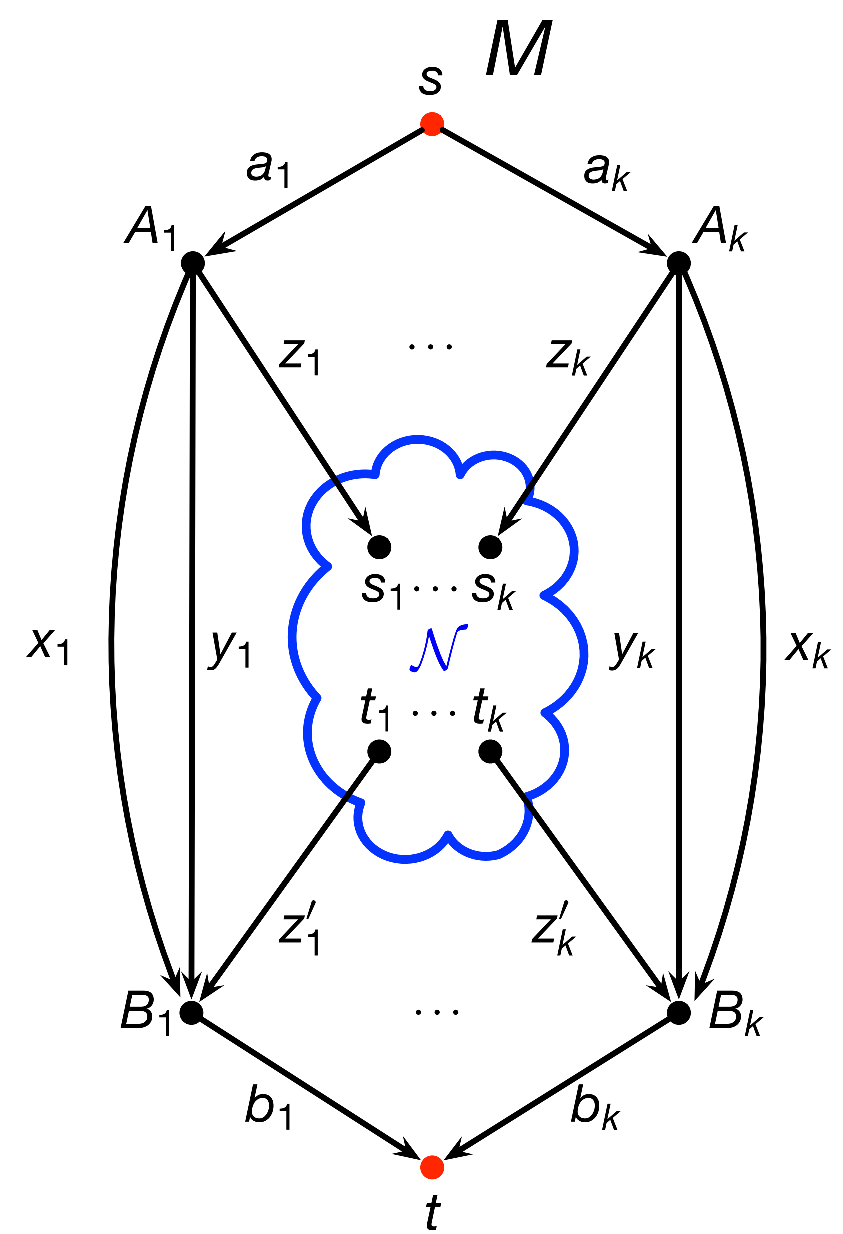
- Assume rate k achievable on $\mathcal{N}'$: show rate $(1, 1, \ldots, 1)$ on $\mathcal{N}$
- Full rate (cut): 1-1 corresp. between message M, $a_1, \ldots, a_k$, $b_1, \ldots, b_k$
- Error correction: For $M_1 \neq M_2$ if $b_i(M_1) \neq b_i(M_2)$ then $z'_i(M_1) \neq z'_i(M_2)$
- For $z'_i(M_1) = z'_i(M_2)$ terminal cannot distinguish between M_1, M_2
- 1-1 correspondence between $b_i \leftrightarrow z'_i$



Zero Error Case

Proof sketch: ⇒

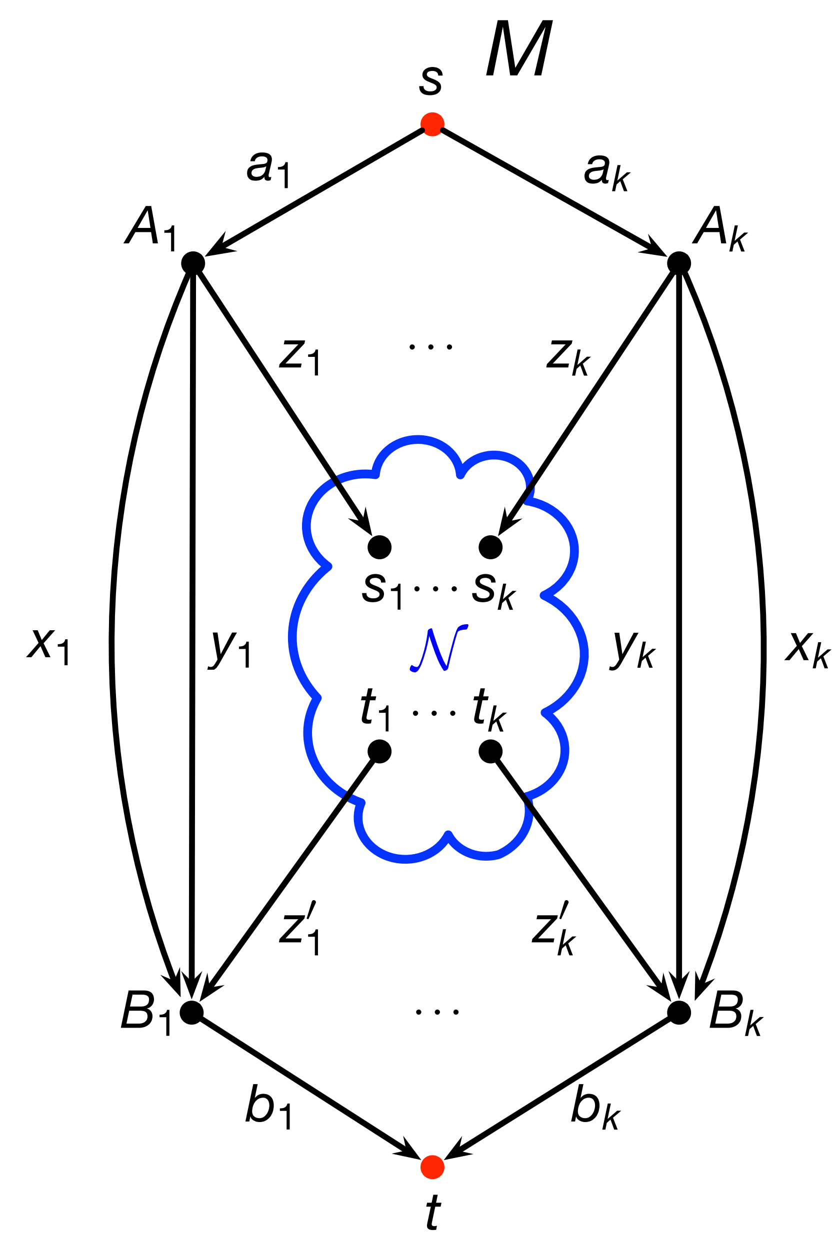
- Assume rate k achievable on $\mathcal{N}'$: show rate $(1, 1, \ldots, 1)$ on $\mathcal{N}$
- Full rate (cut): 1-1 corresp. between message M, $a_1, \ldots, a_k$, $b_1, \ldots, b_k$
- 1-1 correspondence between $b_i \leftrightarrow z_i'$



Zero Error Case

Proof sketch: ➡

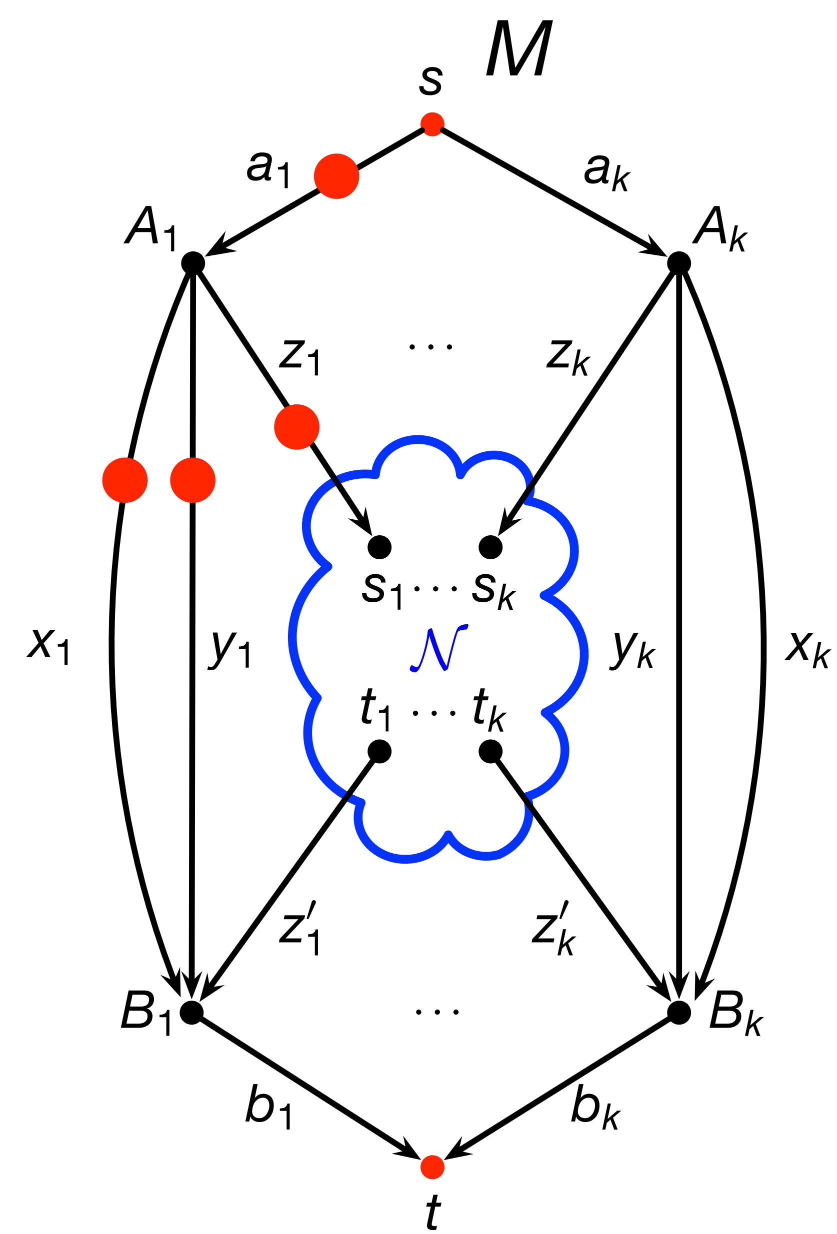
- Assume rate k achievable on $\mathcal{N}'$: show rate $(1, 1, \ldots, 1)$ on $\mathcal{N}$
- Full rate (cut): 1-1 corresp. between message M, $a_1, \ldots, a_k$, $b_1, \ldots, b_k$
- 1-1 correspondence between $b_i \leftrightarrow z_i'$
- Same argument: 1-1 correspondence between $a_i \leftrightarrow x_i \leftrightarrow y_i \leftrightarrow z_i$



Zero Error Case

Proof sketch: ➡

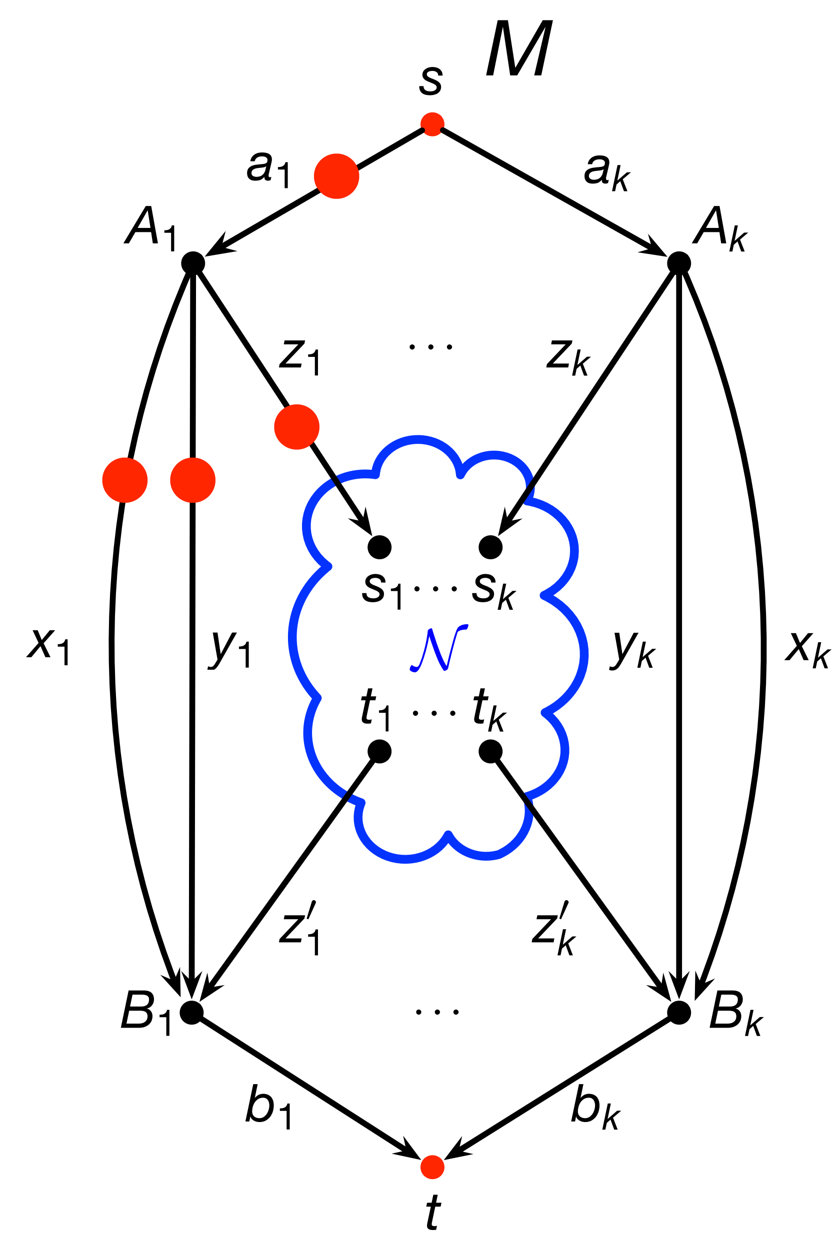
- Assume rate k achievable on $\mathcal{N}'$: show rate $(1, 1, \ldots, 1)$ on $\mathcal{N}$
- Full rate (cut): 1-1 corresp. between message M, $a_1, \ldots, a_k$, $b_1, \ldots, b_k$
- 1-1 correspondence between $b_i \leftrightarrow z_i'$
- Same argument: 1-1 correspondence between $a_i \leftrightarrow x_i \leftrightarrow y_i \leftrightarrow z_i$



Zero Error Case

Proof sketch: ➡

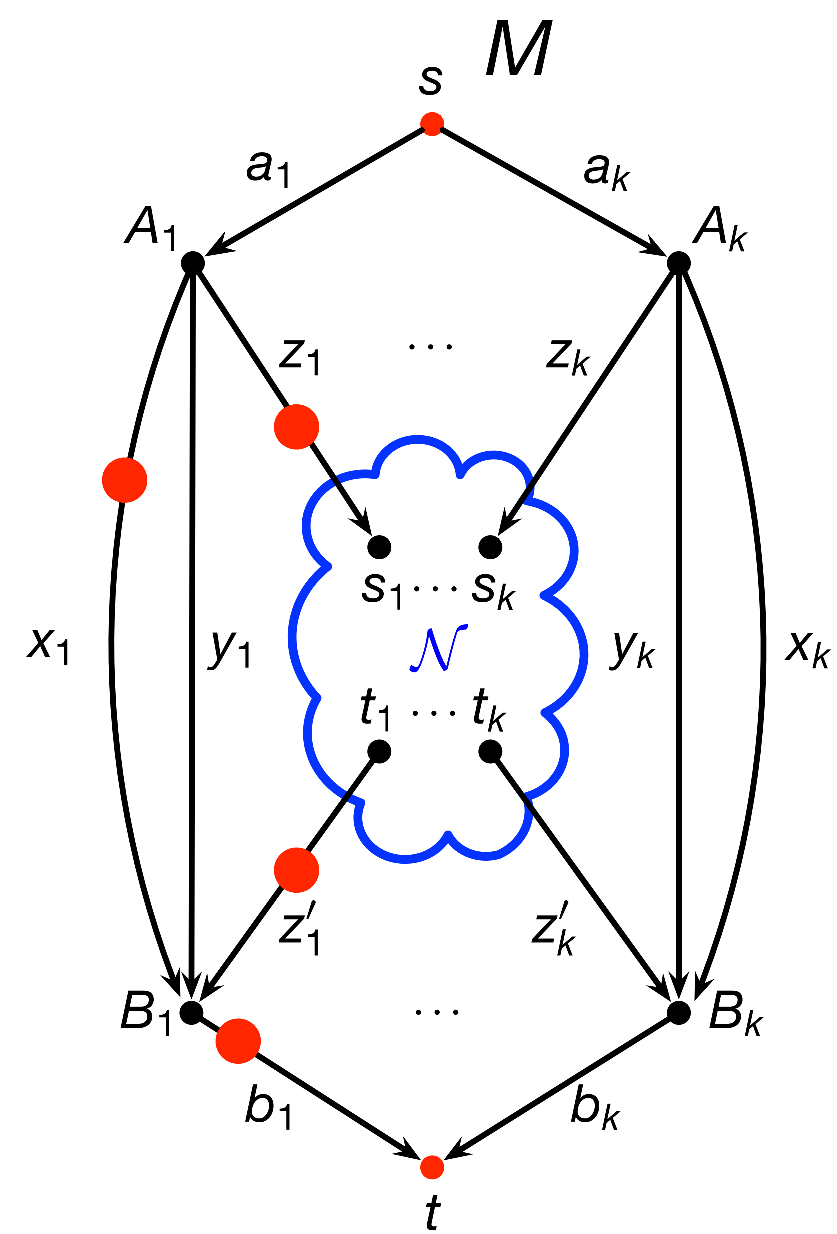
- Assume rate k achievable on $\mathcal{N}'$: show rate $(1, 1, \ldots, 1)$ on $\mathcal{N}$
- Full rate (cut): 1-1 corresp. between message M, $a_1, \ldots, a_k$, $b_1, \ldots, b_k$
- 1-1 correspondence between $b_i \leftrightarrow z_i'$
- Same argument: 1-1 correspondence between $a_i \leftrightarrow x_i \leftrightarrow y_i \leftrightarrow z_i$
- In summary: $z_i \leftrightarrow x_i \leftrightarrow b_i \leftrightarrow z_i'$
- Implies $z_i \leftrightarrow z_i'$: multiple unicast



Zero Error Case

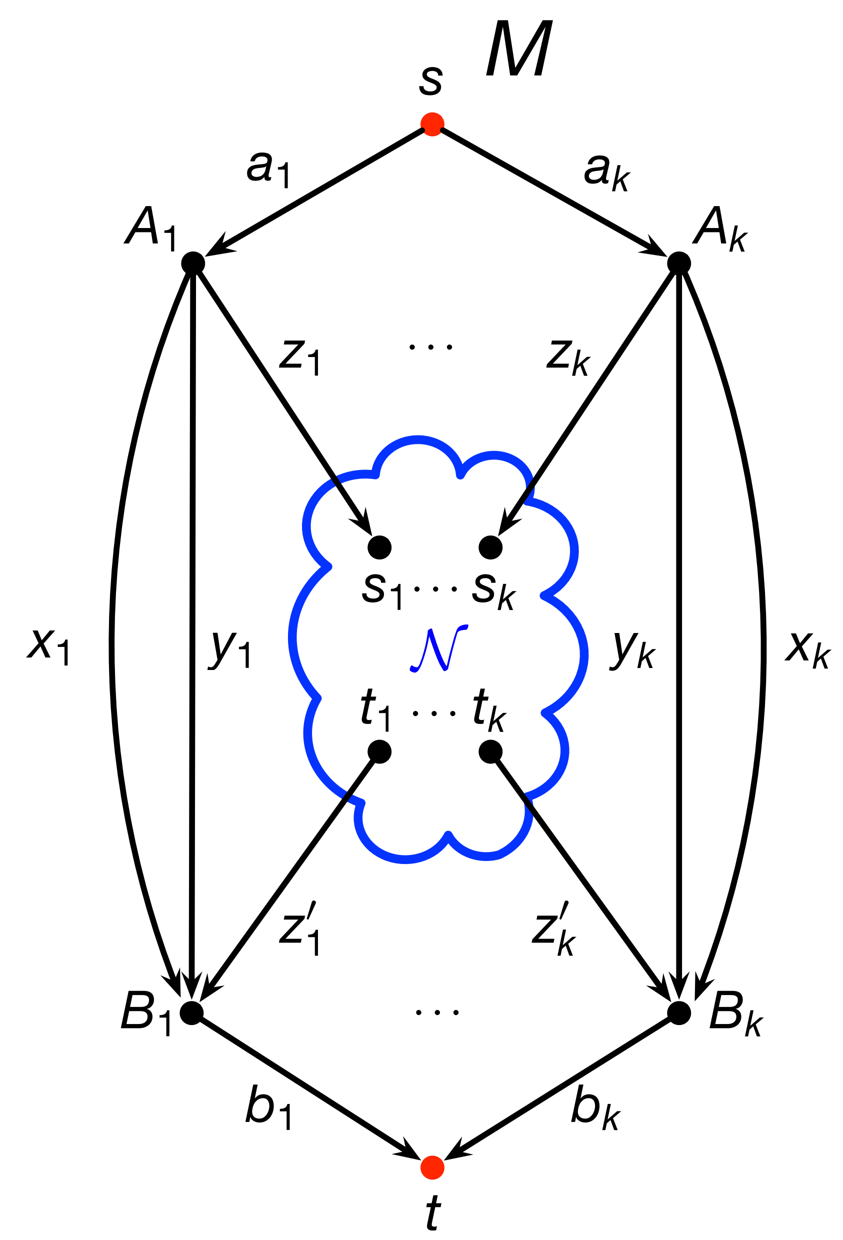
Proof sketch: ⇒

- Assume rate k achievable on $\mathcal{N}'$: show rate $(1, 1, \ldots, 1)$ on $\mathcal{N}$
- Full rate (cut): 1-1 corresp. between message M, $a_1, \ldots, a_k$, $b_1, \ldots, b_k$
- 1-1 correspondence between $b_i \leftrightarrow z'_i$
- Same argument: 1-1 correspondence between $a_i \leftrightarrow x_i \leftrightarrow y_i \leftrightarrow z_i$
- In summary: $z_i \leftrightarrow x_i \leftrightarrow b_i \leftrightarrow z'_i$
- Implies $z_i \leftrightarrow z'_i$: multiple unicast



Asymptotic Error Case

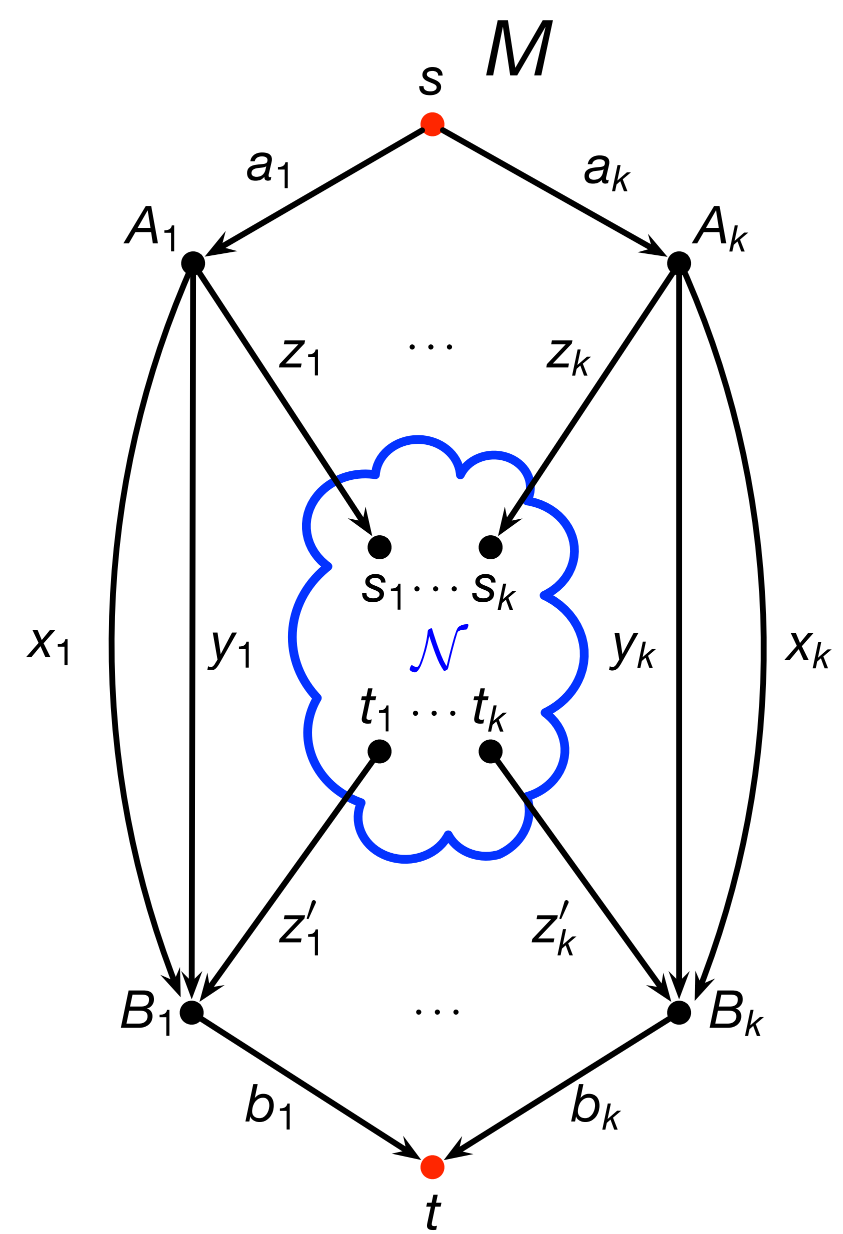
Proof sketch:



Asymptotic Error Case

Proof sketch:

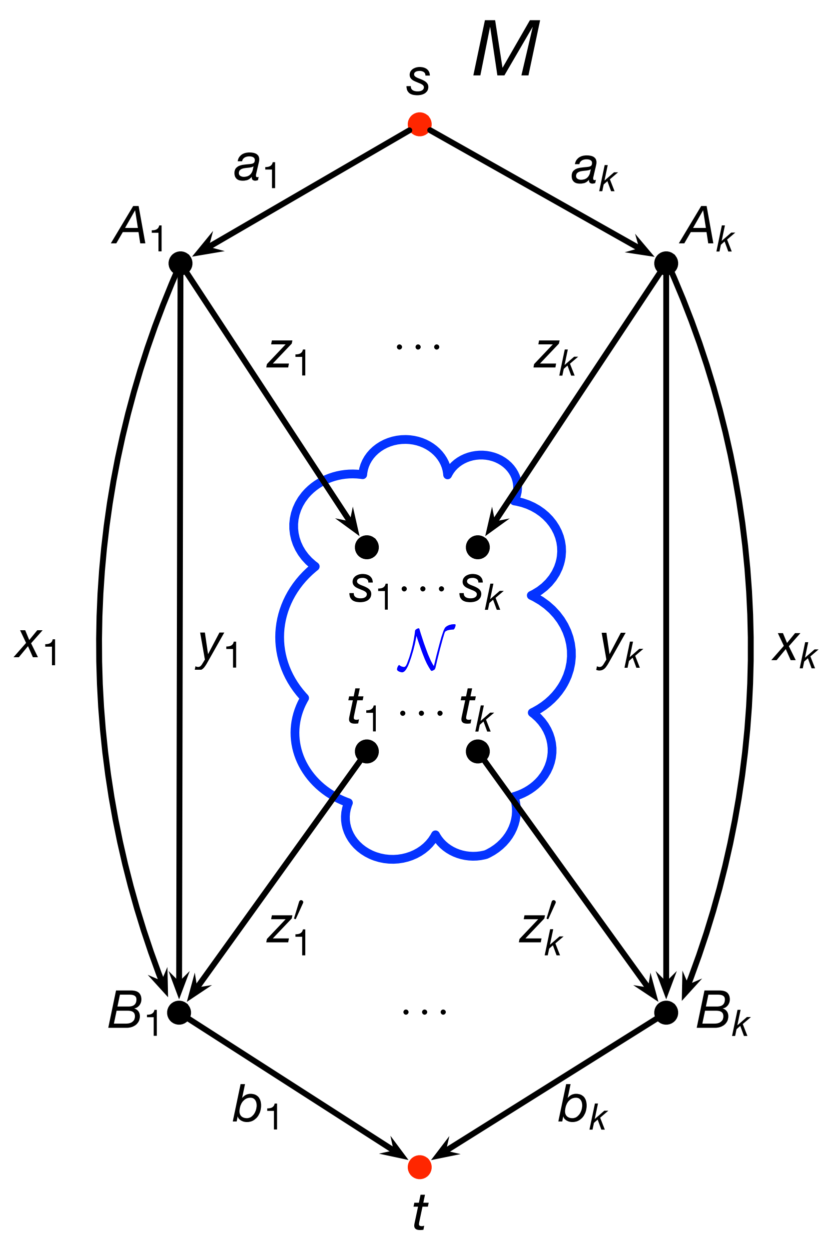
- Arguments are more complicated
- ⇦ Essentially the same



Asymptotic Error Case

Proof sketch:

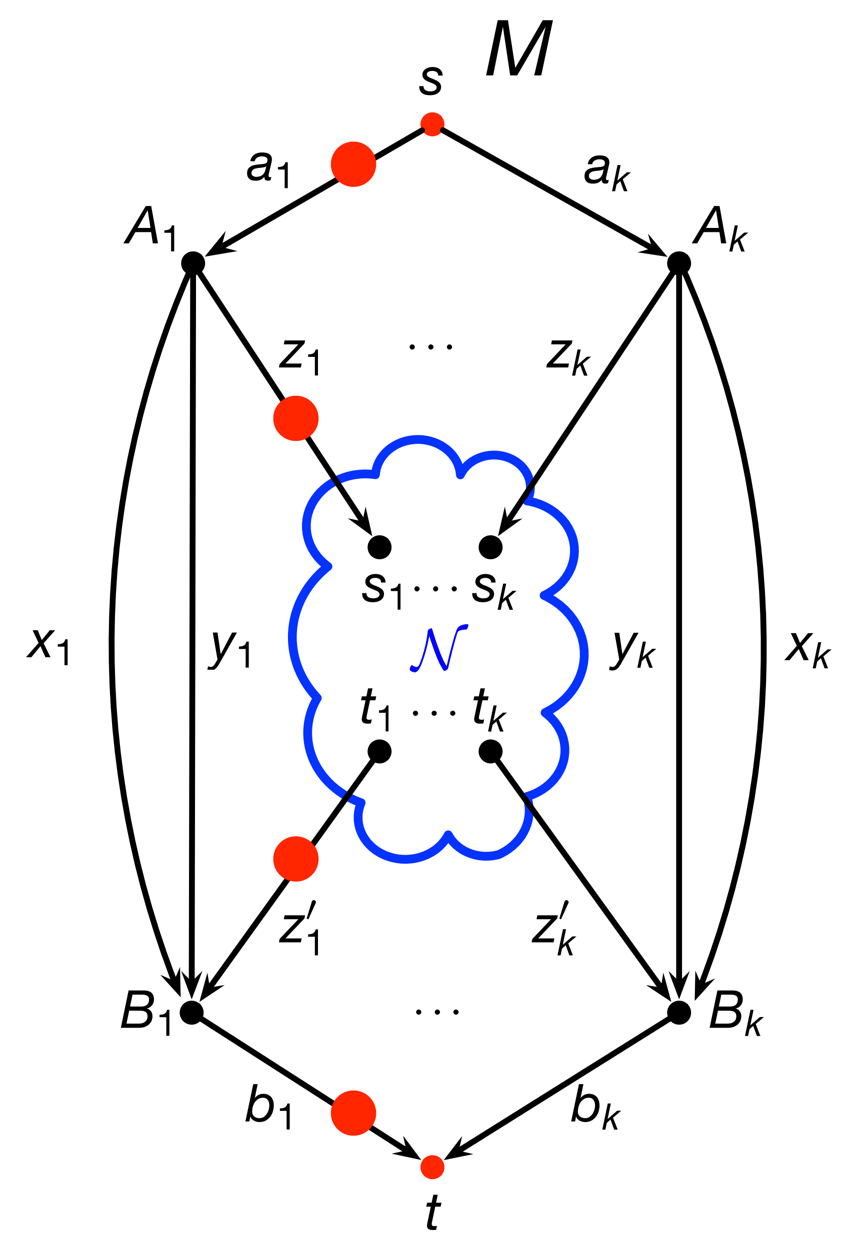
- Arguments are more complicated
- ⇐ Essentially the same
- ⇒ We show:
  - $\lim_{n\to\infty,\epsilon\to 0} I(a_i;b_i)/n = 1$
  - $\lim_{n\to\infty,\epsilon\to 0} I(a_i;z_i)/n = 1$
  - $\lim_{n\to\infty,\epsilon\to 0} I(b_i;z'_i)/n = 1$



Asymptotic Error Case

Proof sketch:

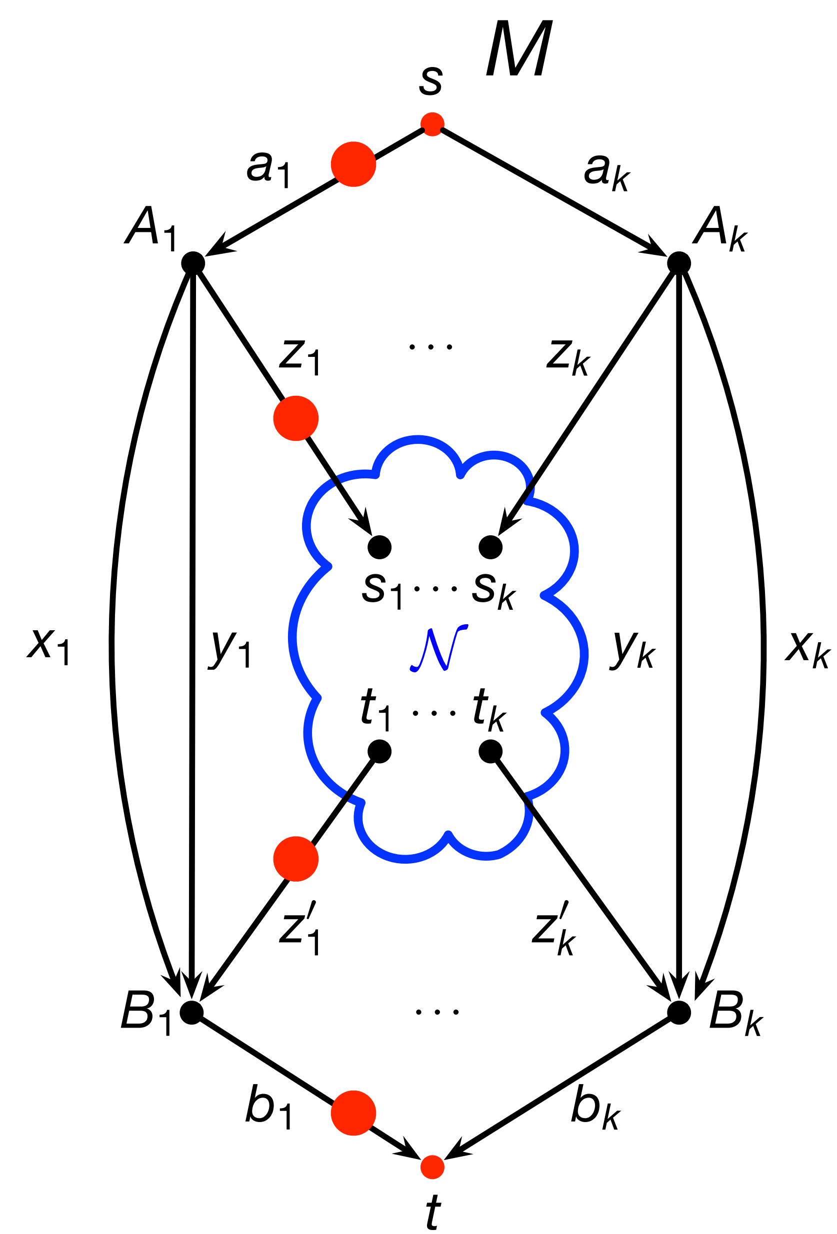
- Arguments are more complicated
- ⇐ Essentially the same
- ⇒ We show:
  - $\lim_{n\to\infty,\epsilon\to 0} I(a_i; b_i)/n = 1$
  - $\lim_{n\to\infty,\epsilon\to 0} I(a_i; z_i)/n = 1$
  - $\lim_{n\to\infty,\epsilon\to 0} I(b_i; z_i')/n = 1$



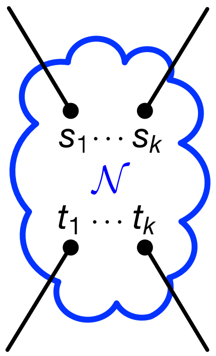
Asymptotic Error Case

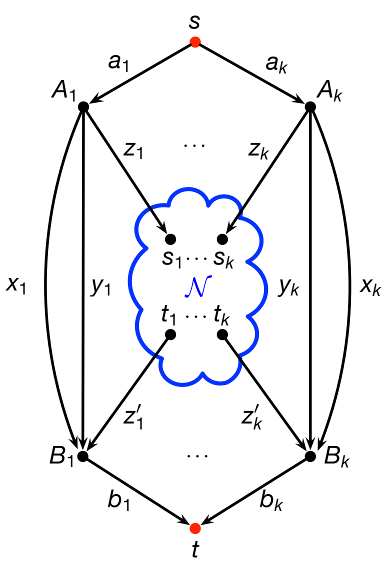
Proof sketch:

- Arguments are more complicated
- ⇐ Essentially the same
- ⇒ We show:
  - $\lim_{n\to\infty,\epsilon\to 0} I(a_i;b_i)/n = 1$
  - $\lim_{n\to\infty,\epsilon\to 0} I(a_i;z_i)/n = 1$
  - $\lim_{n\to\infty,\epsilon\to 0} I(b_i;z_i')/n = 1$
- In summary: $\lim_{n\to\infty,\epsilon\to 0} I(z_i;z_i')/n = 1$
- Block codes: MU with $\epsilon > 0$ possible

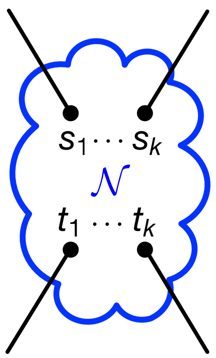


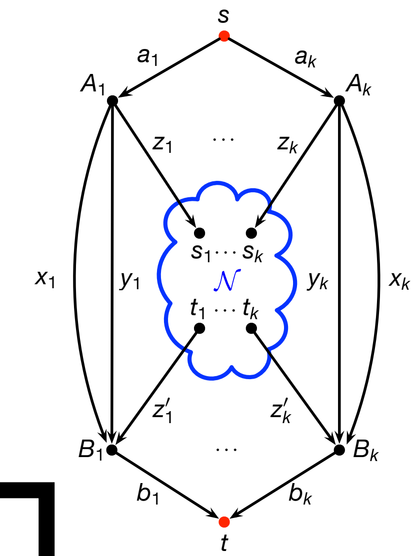
Current Understanding



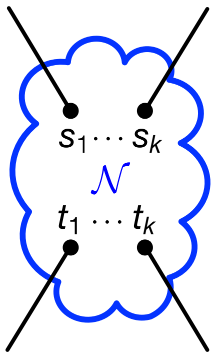


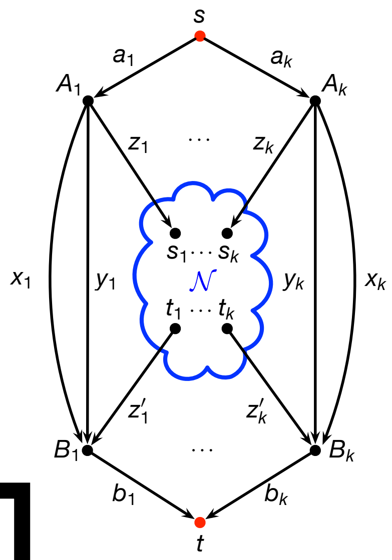
Current Understanding





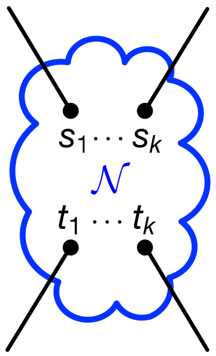
Current Understanding

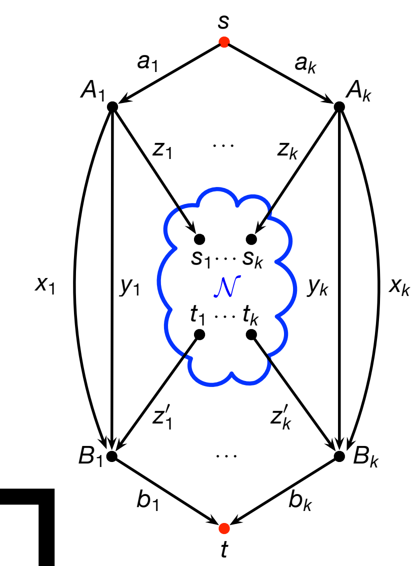




$(1, 1, \ldots, 1)$ feasible with zero error ⟷ Rate k feasible with zero error

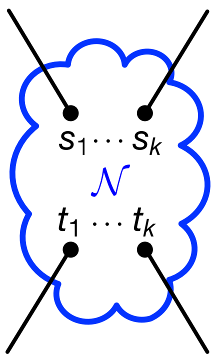
Current Understanding

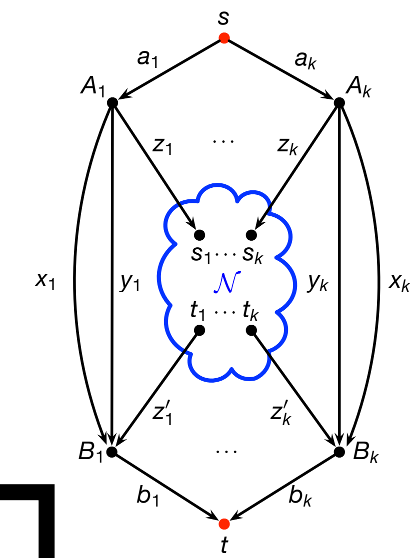




$(1, 1, \ldots, 1)$ feasible with zero error	⟺	Rate k feasible with zero error
$(1, 1, \ldots, 1)$ feasible with ϵ error	⟺	Rate k feasible with ϵ error

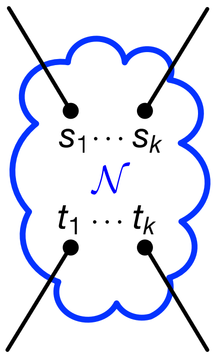
Current Understanding

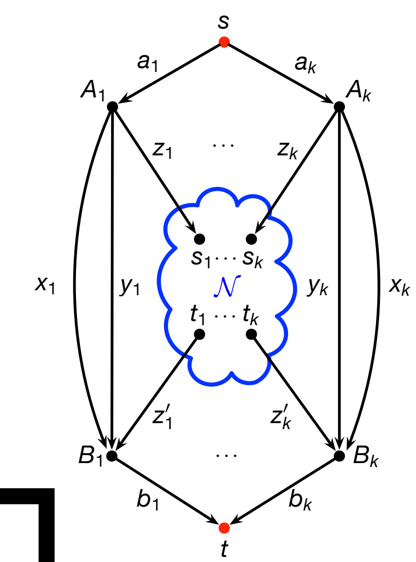




$(1, 1, \ldots, 1)$ feasible with zero error	⟺	Rate k feasible with zero error
$(1, 1, \ldots, 1)$ feasible with ϵ error	⟺	Rate k feasible with ϵ error
$(1, 1, \ldots, 1)$ asymp. feasible, not exactly		Rate k asymp. feasible, not exactly
$(1, 1, \ldots, 1)$ **not** asymp. feasible		Rate k **not** asymp. feasible

Current Understanding





$(1, 1, \ldots, 1)$ feasible with zero error	⟺	Rate *k* feasible with zero error
$(1, 1, \ldots, 1)$ feasible with ϵ error	⟺	Rate *k* feasible with ϵ error
$(1, 1, \ldots, 1)$ asymp. feasible, not exactly	⟺	Rate *k* asymp. feasible, not exactly
$(1, 1, \ldots, 1)$ **not** asymp. feasible	⟺	Rate *k* **not** asymp. feasible

Infeasibility of Multiple Unicast

Theorem

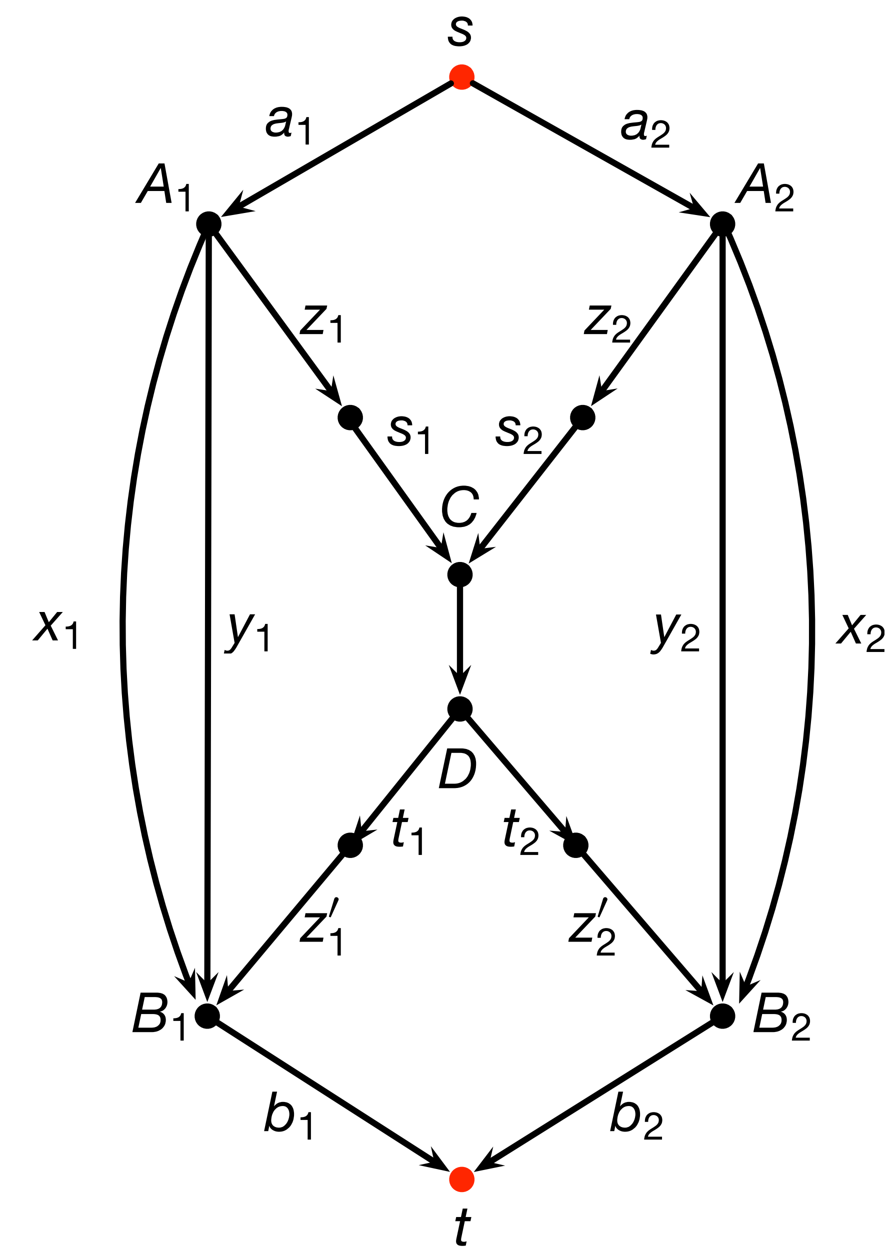
There exist networks $\mathcal{N}$ and $\mathcal{N}'$ such that rate *k* is asymptotically achievable on $\mathcal{N}'$ but $(1, 1, \ldots, 1)$ is not asymp. achievable on $\mathcal{N}$.

Infeasibility of Multiple Unicast

Theorem

There exist networks $\mathcal{N}$ and $\mathcal{N}'$ such that rate k is asymptotically achievable on $\mathcal{N}'$ but $(1, 1, \ldots, 1)$ is not asymp. achievable on $\mathcal{N}$.

Proof (constructive):



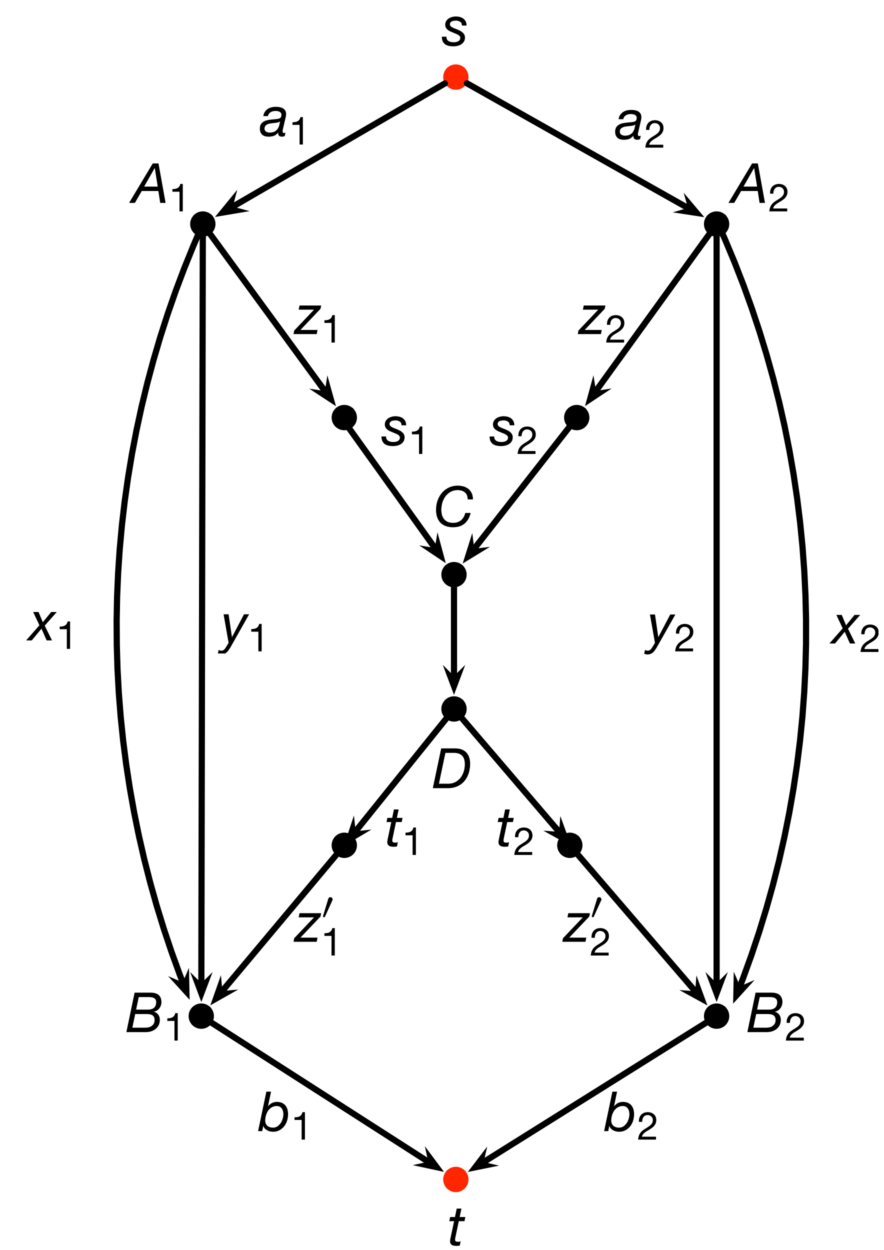
Infeasibility of Multiple Unicast

Theorem

There exist networks $\mathcal{N}$ and $\mathcal{N}'$ such that rate *k* is asymptotically achievable on $\mathcal{N}'$ but $(1, 1, \ldots, 1)$ is not asymp. achievable on $\mathcal{N}$.

Proof (constructive):

- Unit capacity edges, messages $M = (M_1, M_2)$, $H(M_1) = H(M_2) = n$-1 bits, adversary can access one link
- Network code:



Infeasibility of Multiple Unicast

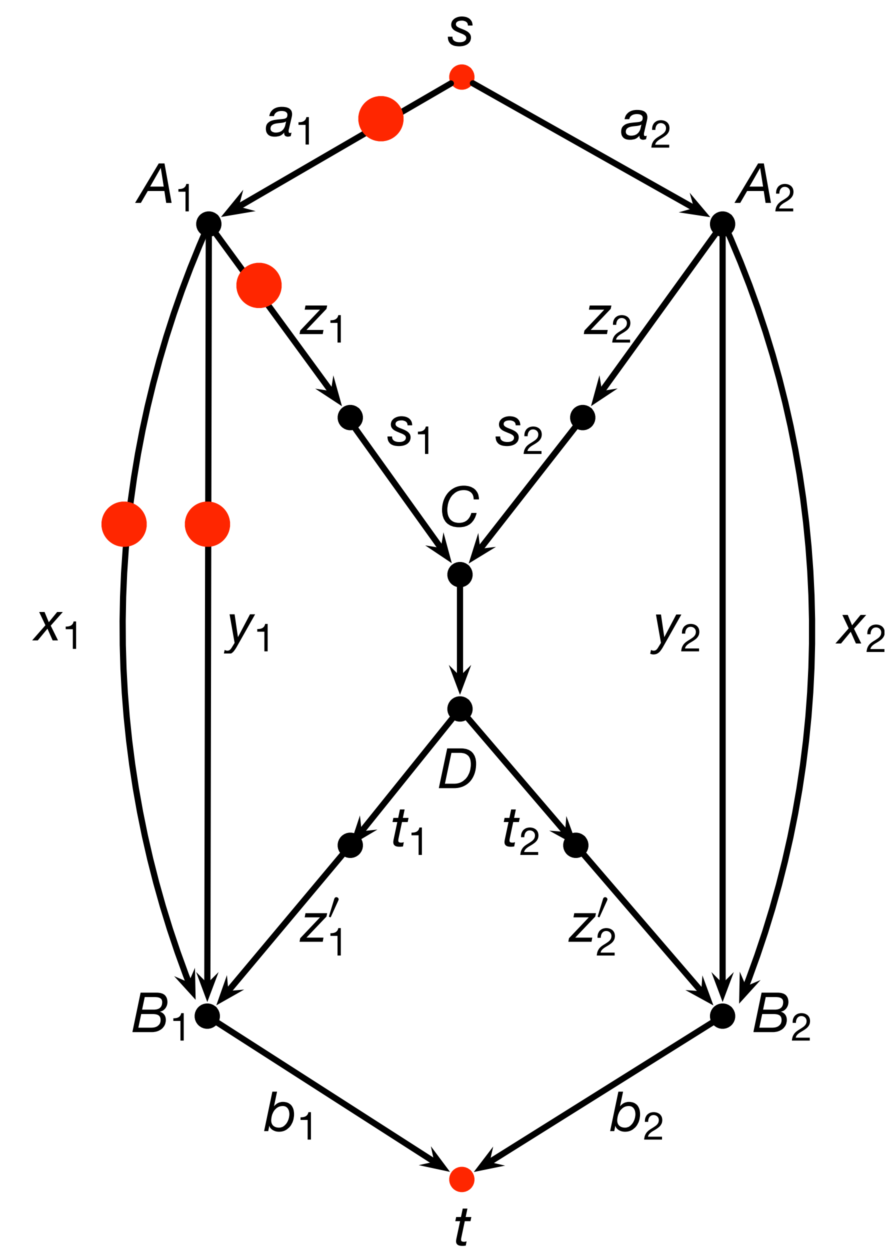
Theorem

There exist networks $\mathcal{N}$ and $\mathcal{N}'$ such that rate k is asymptotically achievable on $\mathcal{N}'$ but $(1, 1, \ldots, 1)$ is not asymp. achievable on $\mathcal{N}$.

Proof (constructive):

- Unit capacity edges, messages $M = (M_1, M_2)$, $H(M_1) = H(M_2) = n\text{-}1$ bits, adversary can access one link

- Network code:
$a_1(M) = x_1(M) = y_1(M) = z_1(M) = M_1$



Infeasibility of Multiple Unicast

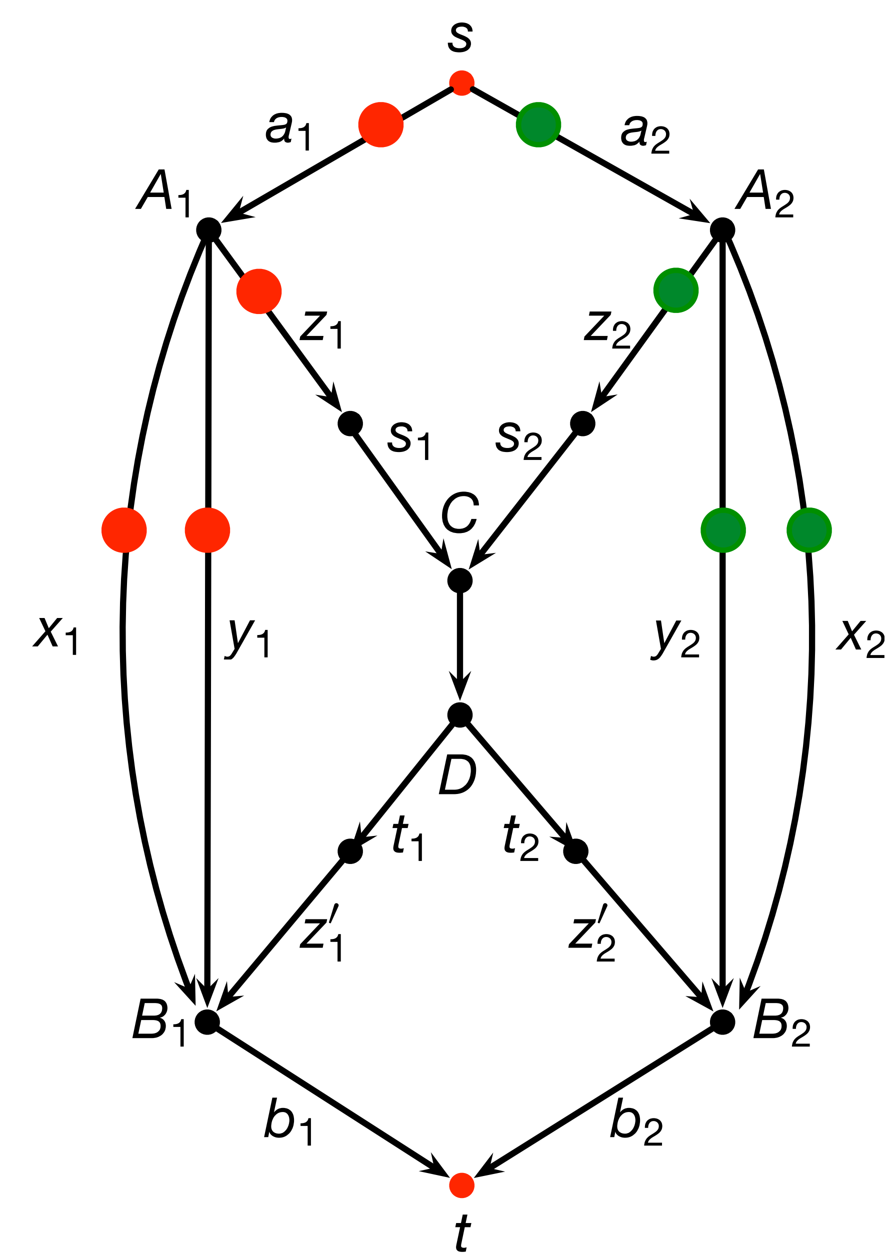
Theorem

There exist networks $\mathcal{N}$ and $\mathcal{N}'$ such that rate k is asymptotically achievable on $\mathcal{N}'$ but $(1, 1, \ldots, 1)$ is not asymp. achievable on $\mathcal{N}$.

Proof (constructive):

- Unit capacity edges, messages $M = (M_1, M_2)$, $H(M_1) = H(M_2) = n\text{-}1$ bits, adversary can access one link
- Network code:

$$a_1(M) = x_1(M) = y_1(M) = z_1(M) = M_1$$
$$a_2(M) = x_2(M) = y_2(M) = z_2(M) = M_2$$



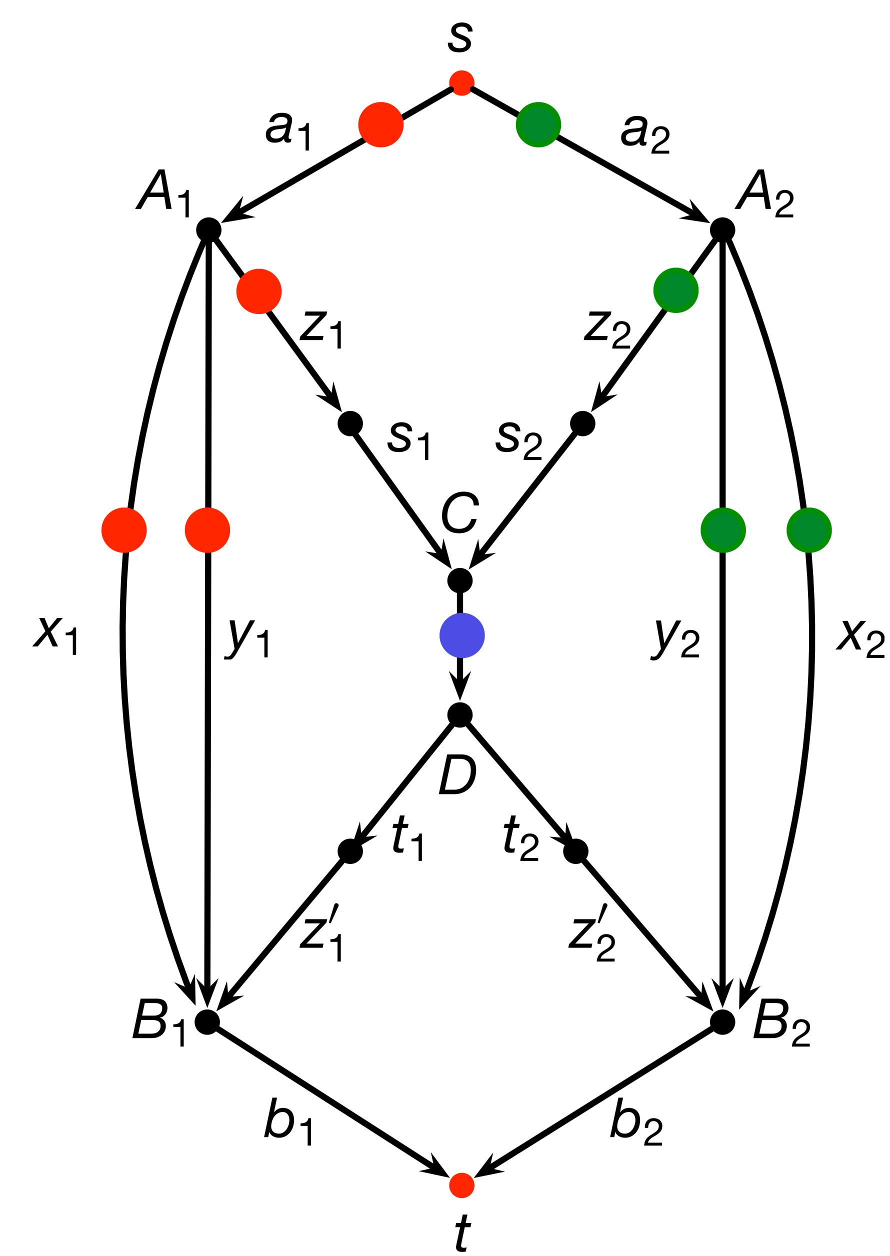
Infeasibility of Multiple Unicast

Theorem

There exist networks $\mathcal{N}$ and $\mathcal{N}'$ such that rate k is asymptotically achievable on $\mathcal{N}'$ but $(1, 1, \ldots, 1)$ is not asymp. achievable on $\mathcal{N}$.

Proof (constructive):

- Unit capacity edges, messages $M = (M_1, M_2)$, $H(M_1) = H(M_2) = n$-1 bits, adversary can access one link
- Network code:
  $a_1(M) = x_1(M) = y_1(M) = z_1(M) = M_1$
  $a_2(M) = x_2(M) = y_2(M) = z_2(M) = M_2$
  $z_1'(M) = z_2'(M) = M_1 + M_2$

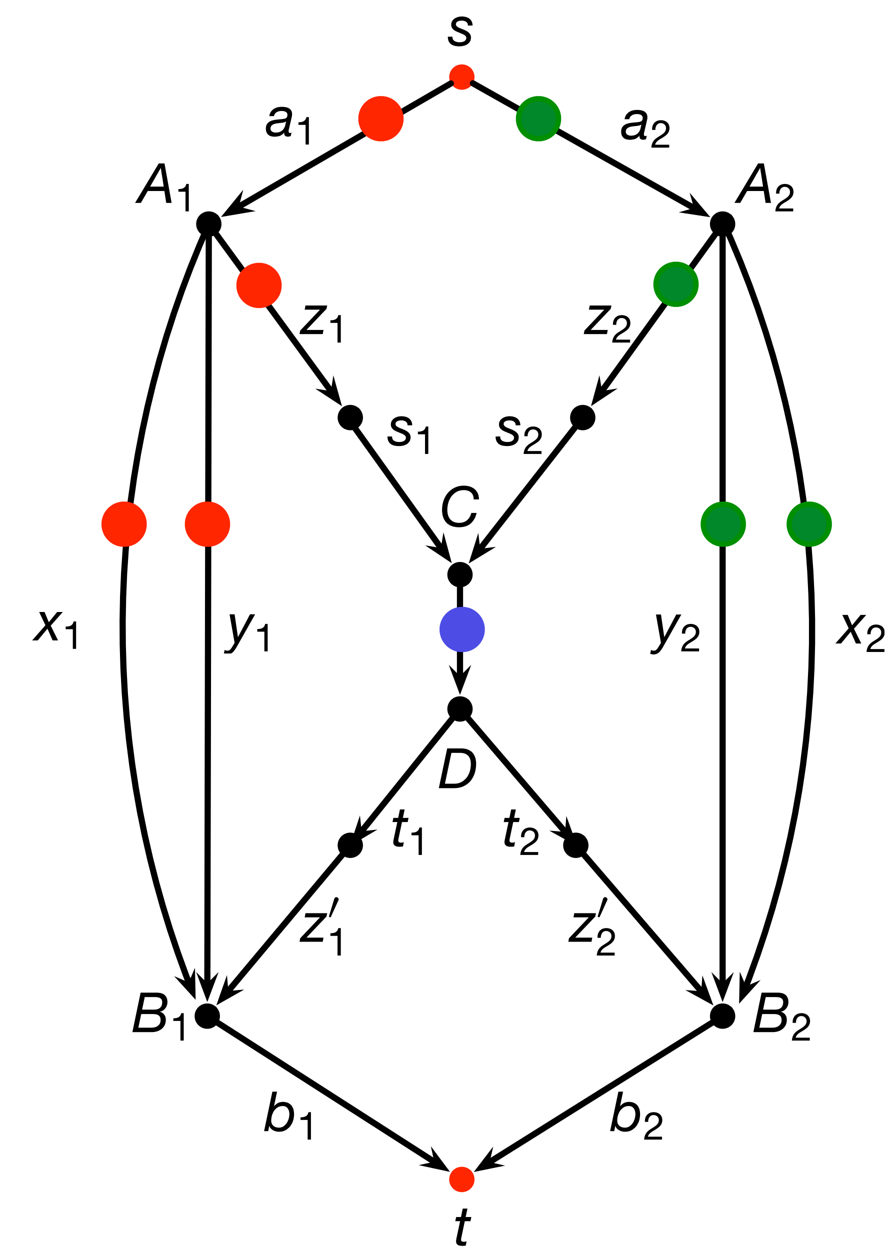


Infeasibility of Multiple Unicast

Theorem

There exist networks $\mathcal{N}$ and $\mathcal{N}'$ such that rate k is asymptotically achievable on $\mathcal{N}'$ but $(1, 1, \ldots, 1)$ is not asymp. achievable on $\mathcal{N}$.

Proof (cont.):



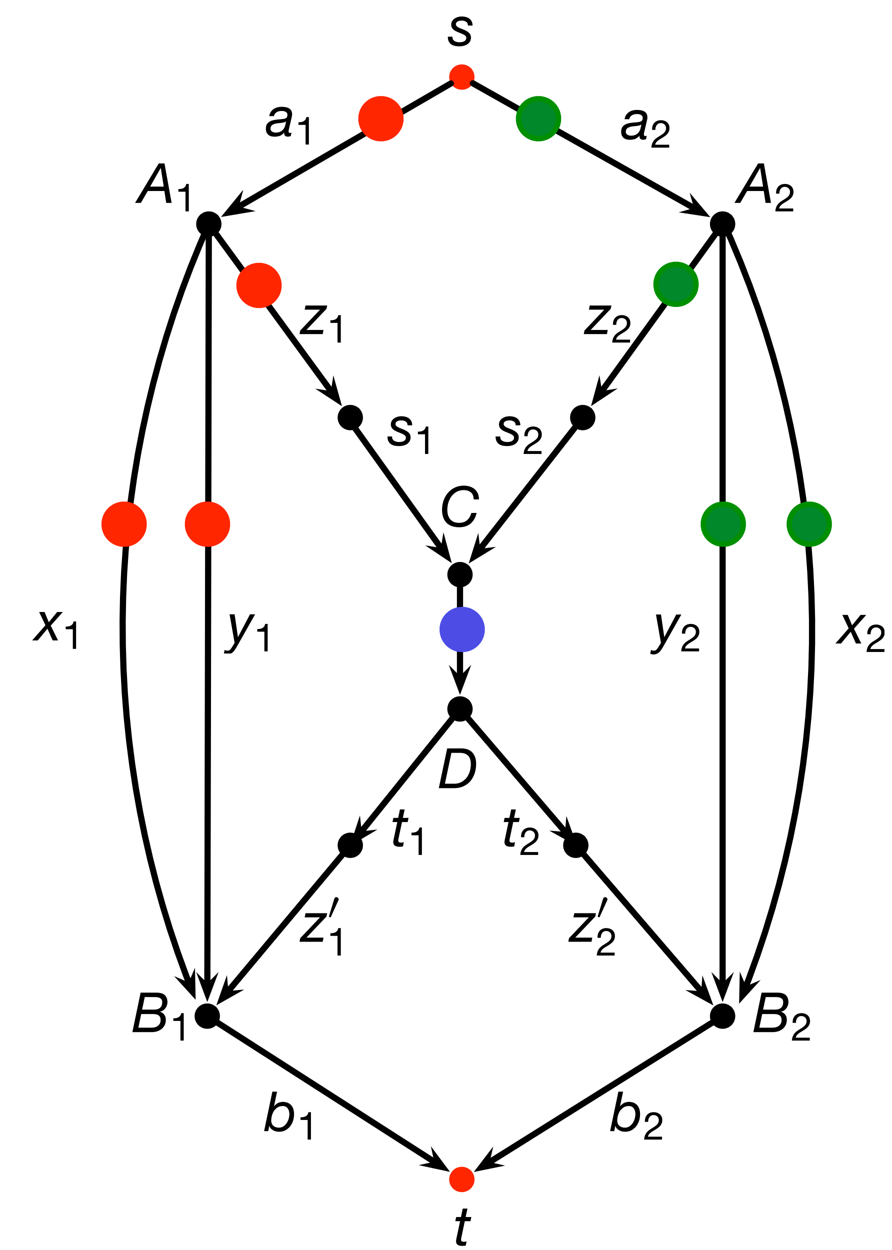
Infeasibility of Multiple Unicast

Theorem

There exist networks $\mathcal{N}$ and $\mathcal{N}'$ such that rate k is asymptotically achievable on $\mathcal{N}'$ but $(1, 1, \ldots, 1)$ is not asymp. achievable on $\mathcal{N}$.

Proof (cont.):

- $b_i(x_i, x_i, z_i') = \begin{cases} x_i & \text{if } x_i = y_i \text{ (case 1)} \\ z_i' & \text{if } x_i \neq y_i \text{ (case 2)} \end{cases}$
- b_i reserves one bit to indicate if case 1 or 2 happens



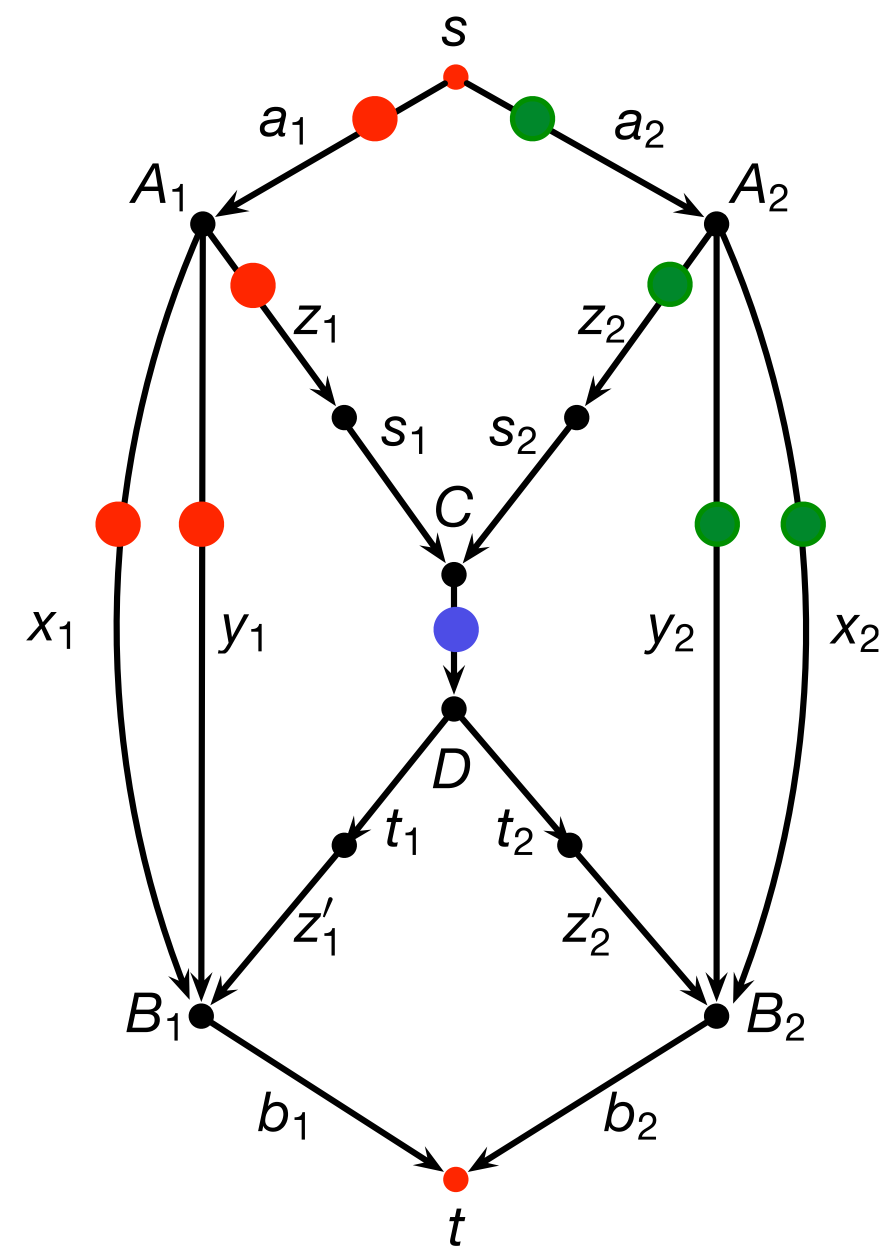
Infeasibility of Multiple Unicast

Theorem

There exist networks $\mathcal{N}$ and $\mathcal{N}'$ such that rate k is asymptotically achievable on $\mathcal{N}'$ but $(1, 1, \ldots, 1)$ is not asymp. achievable on $\mathcal{N}$.

Proof (cont.):

- $b_i(x_i, x_i, z_i') = \begin{cases} x_i & \text{if } x_i = y_i \text{ (case 1)} \\ z_i' & \text{if } x_i \neq y_i \text{ (case 2)} \end{cases}$
- b_i reserves one bit to indicate if case 1 or 2 happens
- t is able to decode (M_1, M_2) correctly at asymptotic rate 2
- Multiple unicast with rate (1, 1) is not feasible



Take Aways

- Error correction in a simple network setting
- Single-source network error correction is at least as hard as multiple-unicast (error-free) network coding
- Falls into broader framework that connects different network communication problems via reductions

Take Aways

- Error correction in a simple network setting
- Single-source network error correction is at least as hard as multiple-unicast (error-free) network coding
- Falls into broader framework that connects different network communication problems via reductions
- [Huang, Langberg, Kliewer, accepted for ISIT 2015]