On Pricing for Routing and Flow-Control in Payment Channel Networks

Suryanarayana Sankagiri[1] Bruce Hajek[2]

[1]School of Computer and Communication Sciences
EPFL, Switzerland

[2]Department of Electrical and Computer Engineering
UIUC, USA

IISc Centre for Networked Intelligence Seminar

Abstract

A payment channel network is a blockchain-based overlay mechanism that allows parties to transact more efficiently than directly using the blockchain. These networks are composed of payment channels that carry transactions between pairs of users. Due to its design, a payment channel cannot sustain a net flow of money in either direction indefinitely. Therefore, a payment channel network cannot serve transaction requests arbitrarily over a long period of time. We give a brief overview of algorithms for pricing and routing in payment channel networks and contrast them with payment bridges. Then we introduce DEBT control, a joint routing and flow-control protocol that guides a payment channel network towards an optimal operating state for any steady-state demand.

Based on joint work with Suryanarayana Sankagiri appearing in IEEE/ACM Transactions on Networking
`https://ieeexplore.ieee.org/document/11096409`

Table of Contents

Table of Contents

Enhancing blockchains

- Bitcoin is limited to 3-7 transactions/second and currently costs around \$0.50 - \$1.00/transaction. Privacy is limited because transaction blocks are public.
- Increasing speed or privacy, lowering cost:
  - payment channel networks (main topic of this talk)
  - sidechannel networks
  - rollups
- Cryptocurrency exchange: bridges

Comparison of Lightning and Liquid

	Lightning payment channel network for Bitcoin	Liquid sidechain network for Bitcoin
Transaction size	micro/small transactions	medium to large transactions
On-ramps	open a payment channel	swap BTC for L-BTC
Off-ramps	close a payment channel or swap out	swap L-BTC for BTC
Settlement speed	seconds	Two minutes
Privacy	strong but monitoring possible	very strong
Custody	Singlesig, hot wallets	single or multisig, hot or cold wallets
Trust model	P2P, trust minimized as long as wallet remains online daily	BTC: L-BTC peg requires trust that 2/3+ of federation functionaries are honest

Rollups

- Rollups take transactions off-chain, process and bundle them
- Developed first for Ethereum (which supports smart contracts), becoming available for Bitcoin
- Rely on security of main chain such as Bitcoin
- Main two types of rollups are optimistic and zero knowledge proofs
- Optimistic rollups rely on aggregators and verifiers which post bonds which can be lost if they break rules
- Zero knowledge proofs rely on proof of correctness by aggregators

Bridges, stable coins, wrapped tokens

- The primary function of bridges is to provide exchange of different types of crypto currency
- Requires providers of liquidity – willing to exchange one type of crypto currency for another.
- At least 70 exist, widely available for top ten cryptocurrencies and wallets
- Stable coins also provide for crypto-currency exchange. For example, USDC stable coins (operated by central authority Tether) are offered on many major blockchains, such as the ERC-20 Etherium token. Offer currency exchange by burn and mint.
- Wrapped tokens facilitate currency exchange by delaying liquidity decisions.

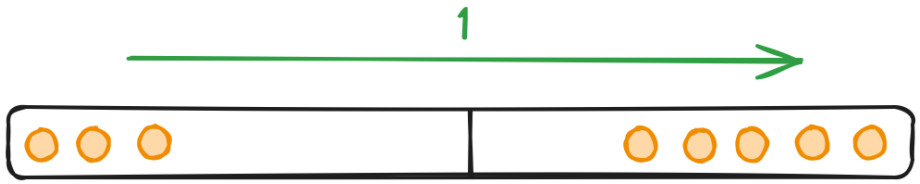
Table of Contents

The Basic Operation of a Payment Channel

- A channel is an escrow fund between two parties
  - The total amount held is called the *channel capacity*
  - The individual amounts are called *channel balances*

The Basic Operation of a Payment Channel

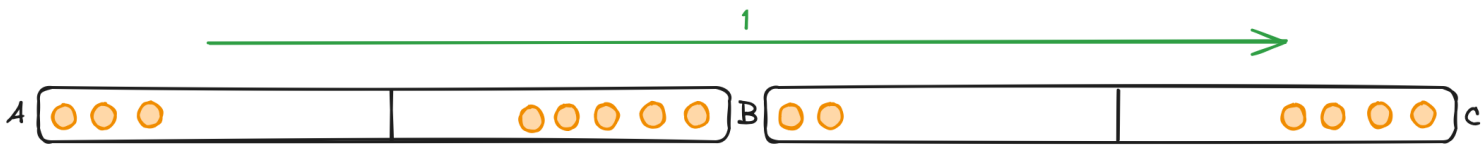
- A channel is an escrow fund between two parties
  - The total amount held is called the *channel capacity*
  - The individual amounts are called *channel balances*



- Parties transact by exchanging messages
  - Balances change while capacity remains constant
  - A channel can carry transactions indefinitely, given sufficient balance

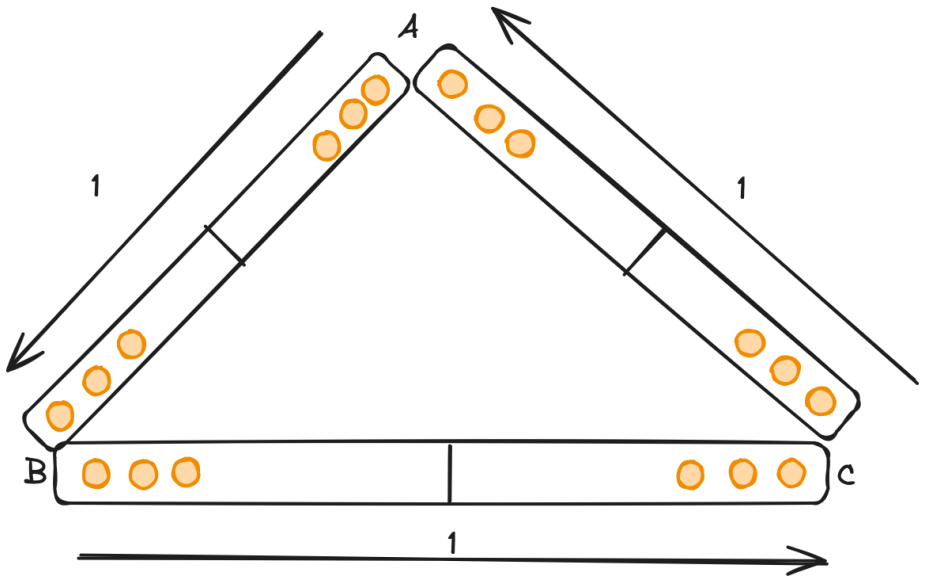
The Basic Operation of a Payment Channel

- A channel is an escrow fund between two parties
  - The total amount held is called the *channel capacity*
  - The individual amounts are called *channel balances*



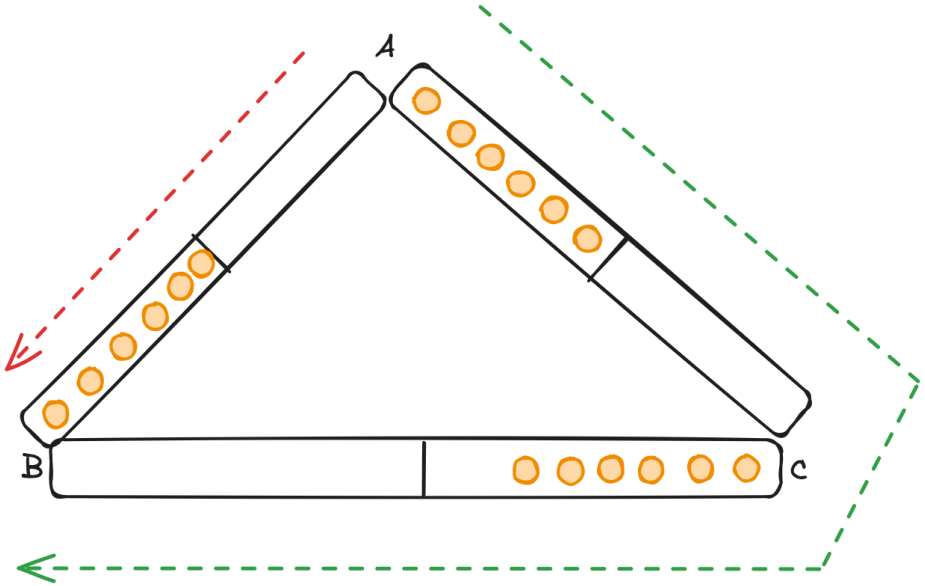
- Parties transact by exchanging messages
  - Balances change while capacity remains constant
  - A channel can carry transactions indefinitely, given sufficient balance
- A transaction can flow through multiple channels
  - Efficient, since channel creation is slow and expensive

The Need for Dynamic Routing



[1] S.M. Varma and S.T. Maguluri. "Throughput optimal routing in blockchain based payment systems". In: *IEEE Transactions on Control of Network Systems* (2021).

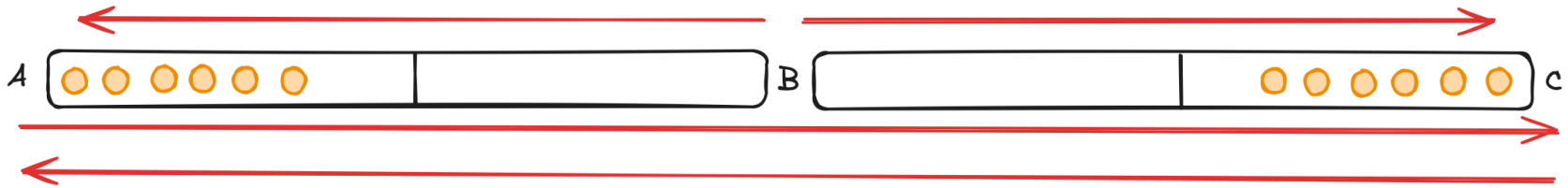
The Need for Dynamic Routing



- Sending money along the shortest path forever is unsustainable
- Switching paths periodically allows nodes to transact forever
- There exist routing schemes[1] that can serve all the demand of a PCN, provided the demand is a *circulation* (net flow from each node is zero)

[1] S.M. Varma and S.T. Maguluri. "Throughput optimal routing in blockchain based payment systems". In: *IEEE Transactions on Control of Network Systems* (2021).

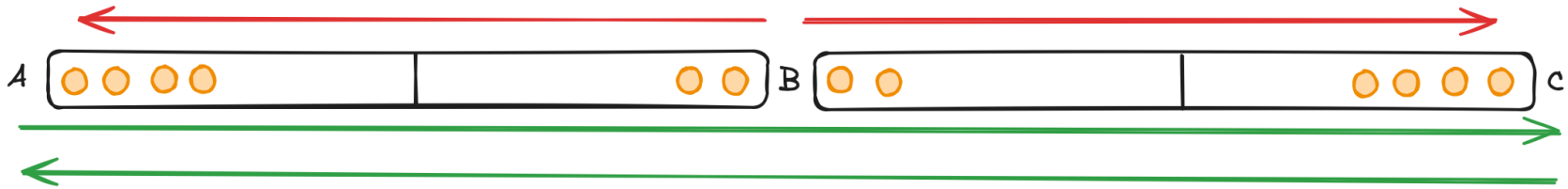
The Need for Flow-Control



- Acyclic demands can lead to *deadlocks*[2], a scenario where the network state blocks circulant flows
- Unlocking a deadlock requires channel resets; slow and expensive

[2] Vibhaalakshmi Sivaraman et al. "The Effect of Network Topology on Credit Network Throughput". In: *Performance Evaluation* (2021).

The Need for Flow-Control



- Acyclic demands can lead to *deadlocks*[2], a scenario where the network state blocks circulant flows
- Unlocking a deadlock requires channel resets; slow and expensive
- Preemptively curtailing some transaction requests can lead to optimal serving of demands

[2] Vibhaalakshmi Sivaraman et al. "The Effect of Network Topology on Credit Network Throughput". In: *Performance Evaluation* (2021).

Table of Contents

Modeling The Network

- The PCN is a graph $G = (V, E)$; $V \rightarrow$ users, $E \rightarrow$ channels
- Channel (u, v) has capacity $c_{u,v}$

Modeling The Network

- The PCN is a graph $G = (V, E)$; $V \rightarrow$ users, $E \rightarrow$ channels
- Channel (u, v) has capacity $c_{u,v}$
- Every node pair (i, j) has a fixed set of paths to transact along:

$$P_{i,j} = \{p_{i,j,k} : k = 1, 2, \ldots, |P_{i,j}|\}$$

- Path information is captured by a routing matrix $R \in \{-1, 0, 1\}^{P \times E}$;

$$P = \cup_{i,j} P_{i,j}$$

Modeling the Demand

- Transaction requests arrive in discrete time slots, at constant rate
  - In each slot t, there is a transaction request of amount $a_{i,j}$ from source i to destination j

$$\mathcal{N} = \{(i,j) : a_{i,j} > 0\}$$

Modeling the Demand

- Transaction requests arrive in discrete time slots, at constant rate
  - In each slot t, there is a transaction request of amount $a_{i,j}$ from source i to destination j

$$\mathcal{N} = \{(i,j) : a_{i,j} > 0\}$$

- The transaction demand is elastic. It is acceptable that the request is dropped or partially served
  - We assume (i,j) gains a utility of $U_{i,j}(f_{i,j})$ upon being served a transaction of amount $f_{i,j} \in [0, a_{i,j}]$
  - We assume $U_{i,j}(\cdot)$ is a concave, differentiable, and nondecreasing function over $[0, a_{i,j}]$, $U_{i,j}(0) = 0$, and $U'_{i,j}(0) < \infty$

Network Flows

Network Flow

With every path in the network, we associate a flow, which represents the amount of money sent along that path over a period of time.

- $f_{i,j,k}[t]$ is the amount of money being sent on path $p_{i,j,k}$ in slot t
- $f_{i,j}[t] = \Sigma_k f_{i,j,k}[t]$ is the total amount of money sent from i to j
- $f[t] \in \mathbb{R}^P$ denotes the set of all flows in the network in slot t
- $f \in \mathbb{R}^P$ denotes a stationary flow

Network State and State-Change Equations

Network State

The state of the network is described by the channel balances.

- $x_{u,v}[t]$ denotes the balance of u in channel (u, v); convention $u < v$
- $x[t] \in \mathbb{R}^E$ denotes the state vector in slot t
- At the end of each slot, the state is updated as $x[t+1] = x[t] - Rf[t]$

Network State and State-Change Equations

Network State

The state of the network is described by the channel balances.

- $x_{u,v}[t]$ denotes the balance of u in channel (u, v); convention $u < v$
- $x[t] \in \mathbb{R}^E$ denotes the state vector in slot t
- At the end of each slot, the state is updated as $x[t+1] = x[t] - Rf[t]$

Feasibility

A flow vector $f[t]$ is feasible (w.r.t. $x[t]$) iff $0 \leq x[t] - Rf[t] \leq c$.
Routing an infeasible flow involves channel resets; expensive and slow.

Our choice of objective

Network Objective

Obtain a stationary flow f^* that:

- maximizes the utility of all users in the network
- can be sustained indefinitely without perpetually resetting any channel

Our choice of objective

Network Objective

Obtain a stationary flow f^* that:

- maximizes the utility of all users in the network
- can be sustained indefinitely without perpetually resetting any channel

- The total utility of the network is $U(f) = \sum_{(i,j)\in\mathcal{N}} U_{i,j}(f_{i,j})$
- The set of *feasible flows* is $A \triangleq \{f : f \geq 0,\ f_{i,j} \leq a_{i,j}\ \forall\ (i,j) \in \mathcal{N}\}$
- f is sustainable iff it satisfies the *detailed balance constraint*: $Rf = 0$

Network Utility Maximization Problem

$$f^* = \arg\max_{f\in A} \quad U(f) \quad \text{subject to} \quad Rf = 0$$

Table of Contents

The Primal and its Dual Problem

$$\max_{f \in A} \quad U(f) \quad \text{such that} \quad Rf = 0 \qquad \text{(Primal)}$$

Let $\lambda_{u,v}$ denote the Lagrange multiplier for the constraint $(Rf)_{u,v} = 0$.

The primal problem, in terms of the Lagrange multipliers, is:

$$\max_{f \in A} \inf_{\lambda \in \mathbb{R}^E} \quad L(f, \lambda); \qquad L(f, \lambda) \triangleq U(f) - \lambda^T Rf$$

The Primal and its Dual Problem

$$\max_{f \in A} \quad U(f) \quad \text{such that} \quad Rf = 0 \qquad \text{(Primal)}$$

Let $\lambda_{u,v}$ denote the Lagrange multiplier for the constraint $(Rf)_{u,v} = 0$.

The primal problem, in terms of the Lagrange multipliers, is:

$$\max_{f \in A} \inf_{\lambda \in \mathbb{R}^E} \quad L(f, \lambda); \qquad L(f, \lambda) \triangleq U(f) - \lambda^T Rf$$

The dual problem is obtained by swapping the min and the max:

$$\inf_{\lambda \in \mathbb{R}^E} \max_{f \in A} \quad L(f, \lambda) \equiv \inf_{\lambda \in \mathbb{R}^E} D(\lambda); \quad D(\lambda) = \max_{f \in A} L(f, \lambda) \qquad \text{(Dual)}$$

$D(\lambda)$ is a convex function and the dual problem is unconstrained

The Gradient of the Dual

The dual problem can be solved by the gradient descent method

Lemma (Corollary of Danskin's Theorem)

The subdifferential set of $D(\lambda)$ is

$$\partial D(\lambda) = \{-Rf : f \in F(\lambda)\}; \quad F(\lambda) = \arg\max_{f \in A} L(f, \lambda)$$

The Gradient of the Dual

The dual problem can be solved by the gradient descent method

Lemma (Corollary of Danskin's Theorem)

The subdifferential set of $D(\lambda)$ is

$$\partial D(\lambda) = \{-Rf : f \in F(\lambda)\}; \quad F(\lambda) = \arg\max_{f \in A} L(f, \lambda)$$

Gradient Descent on $D(\lambda)$

Assuming uniqueness of $F(\lambda)$,

$$\begin{aligned} f[t] &= F(\lambda[t]) \\ \lambda[t+1] &= \lambda[t] + \gamma Rf[t]; \quad \gamma > 0 \end{aligned} \tag{A}$$

From Gradient Descent to Network Protocol

Lagrange Multipliers as Prices

Interpret $\lambda_{u,v}[t]$ as the *channel price*, declared at the beginning of the slot.
Flows respond to the prices: $f[t] = \arg\max_{f \in A} L(f, \lambda[t])$

From Gradient Descent to Network Protocol

Lagrange Multipliers as Prices

Interpret $\lambda_{u,v}[t]$ as the *channel price*, declared at the beginning of the slot. Flows respond to the prices: $f[t] = \arg\max_{f \in A} L(f, \lambda[t])$

Flow Computation

$$f[t] = \arg\max_{f \in A} U(f) - \lambda[t]^T Rf$$

$$\mu = R^T \lambda \Rightarrow U(f) - \mu^T f = \sum_{(i,j) \in \mathcal{N}} \left(U_{i,j}(f_{i,j}) - \sum_k f_{i,j,k} \mu_{i,j,k} \right)$$

$$A \equiv f_{i,j,k} \geq 0, f_{i,j} = \sum_k f_{i,j,k} \leq a_{i,j} \ \forall \ (i,j) \in \mathcal{N}$$

$\mu_{i,j,k}$ is the *path price* for the path $p_{i,j,k} \Rightarrow \sum_k f_{i,j,k} \mu_{i,j,k}$ is the total cost of routing a transaction of value $f_{i,j}$ over different paths.

From Gradient Descent to Network Protocol

Joint Routing and Flow-Control

Each node pair solves the following optimization problem simultaneously:

$$\max_{f_{i,j,k} \geq 0, f_{i,j} \leq a_{i,j}} U_{i,j}(f_{i,j}) - \sum_k f_{i,j,k} \mu_{i,j,k}$$

- Choose the path with the minimum path price (**routing**), and
- Set the amount $f_{i,j} = {U'_{i,j}}^{-1}(\mu_{i,j})$ (**flow control**)
- Multiple solutions possible if two paths have minimum price
- Adding a regularizer $-\eta_{i,j} \sum_k (f_{i,j,k})^2$ incentivizes balanced flows

From Gradient Descent to Network Protocol

Joint Routing and Flow-Control

Each node pair solves the following optimization problem simultaneously:

$$\max_{f_{i,j,k} \geq 0, f_{i,j} \leq a_{i,j}} U_{i,j}(f_{i,j}) - \sum_k f_{i,j,k} \mu_{i,j,k}$$

- Choose the path with the minimum path price (**routing**), and
- Set the amount $f_{i,j} = {U'_{i,j}}^{-1}(\mu_{i,j})$ (**flow control**)
- Multiple solutions possible if two paths have minimum price
- Adding a regularizer $-\eta_{i,j} \sum_k (f_{i,j,k})^2$ incentivizes balanced flows

Prices Reflect State

Prices updated after transactions executed: $\lambda[t+1] = \lambda[t] + \gamma R f[t]$.
Each channel's price differential proportional to net flow through it.
Price $\lambda[t]$ proportional to $x[t] - x[0]$ (provided no resets).

Table of Contents

Convergence of Gradient Descent

Proposition

Assuming $\eta_{i,j} \geq \eta > 0$ *and* $\gamma \leq \|R\|_{op}^2/\eta$*:*

- $D(\lambda[t]) - D(\lambda^*) \leq \frac{\|\lambda^*\|}{2\gamma t} \ \forall \ t \geq 1.$
- $\lambda[t] \to \lambda^{**}$ *for* $\lambda^{**} \in \arg\min_{\lambda \in \mathbb{R}^E} D(\lambda)$ *as* $t \to \infty$.
- $f[t] \xrightarrow{t \to \infty} f^*$*, where* f^* *is the unique solution to the primal problem.*

Convergence of Gradient Descent

Proposition

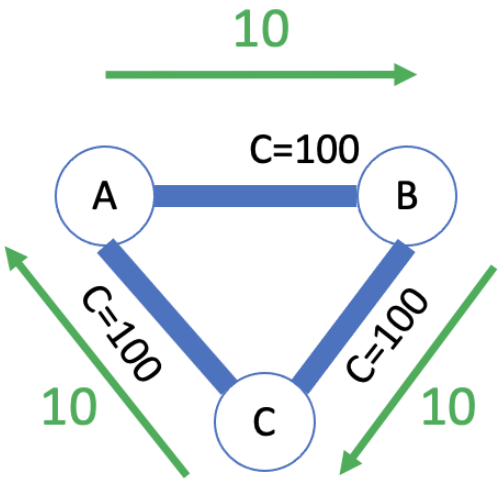
Assuming $\eta_{i,j} \geq \eta > 0$ *and* $\gamma \leq \|R\|_{op}^2/\eta$*:*

- $D(\lambda[t]) - D(\lambda^*) \leq \frac{\|\lambda^*\|}{2\gamma t} \ \forall \ t \geq 1.$
- $\lambda[t] \to \lambda^{**}$ *for* $\lambda^{**} \in \arg\min_{\lambda \in \mathbb{R}^E} D(\lambda)$ *as* $t \to \infty.$
- $f[t] \xrightarrow{t \to \infty} f^*$*, where* f^* *is the unique solution to the primal problem.*

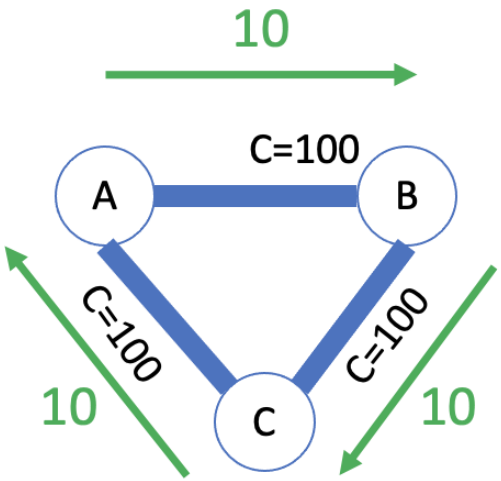
We prove this result by establishing the following facts:

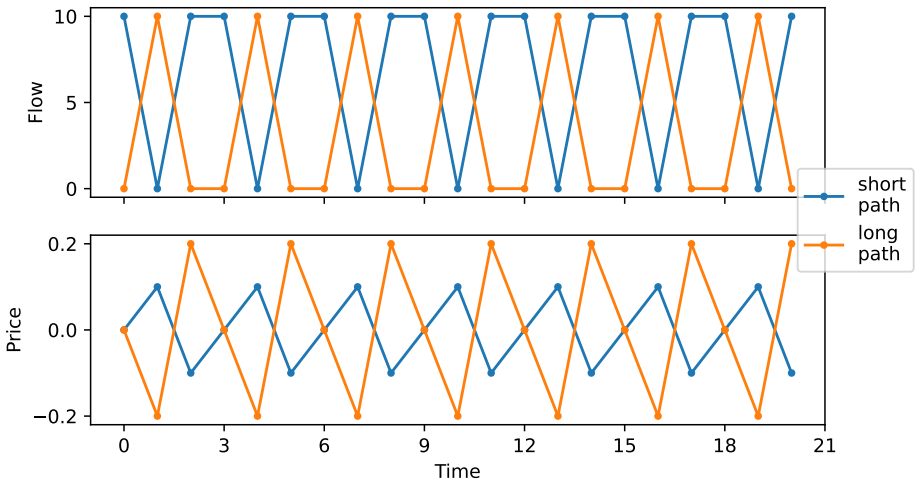
- Strong duality holds between the primal and dual problem
- The dual problem has a finite solution
- The dual function is smooth

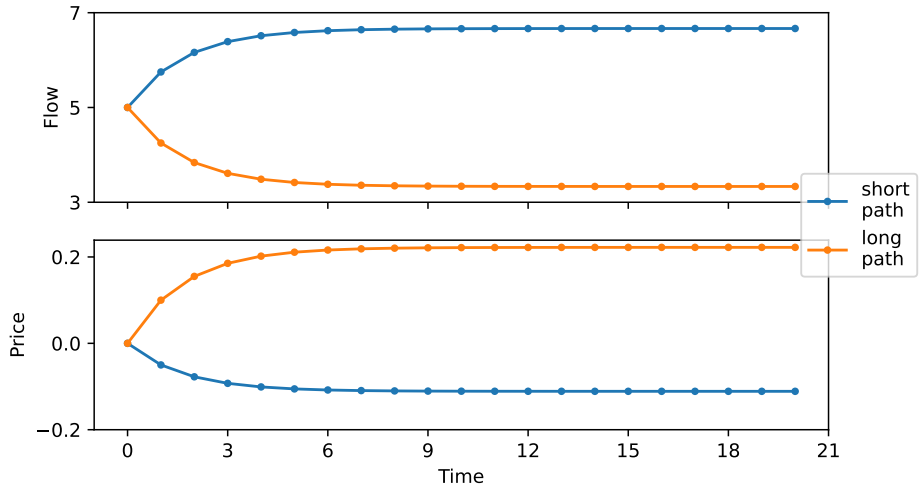
Simulations for cyclic example



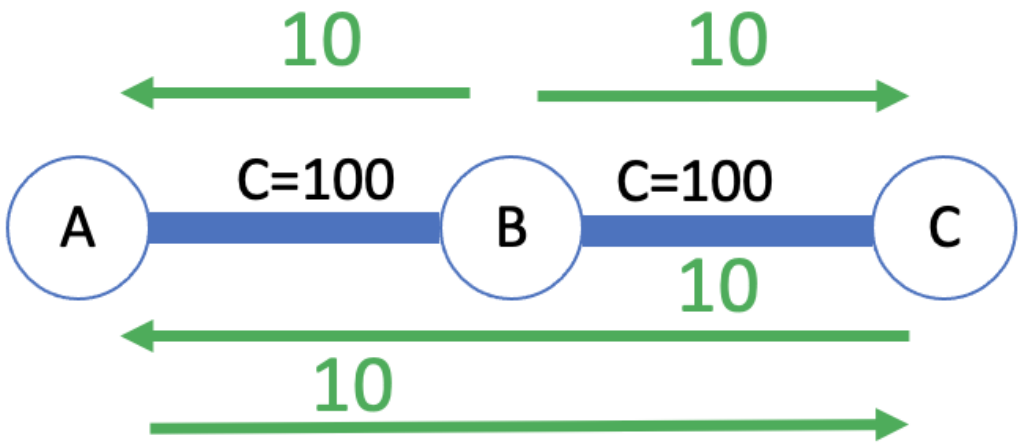
Simulations for cyclic example



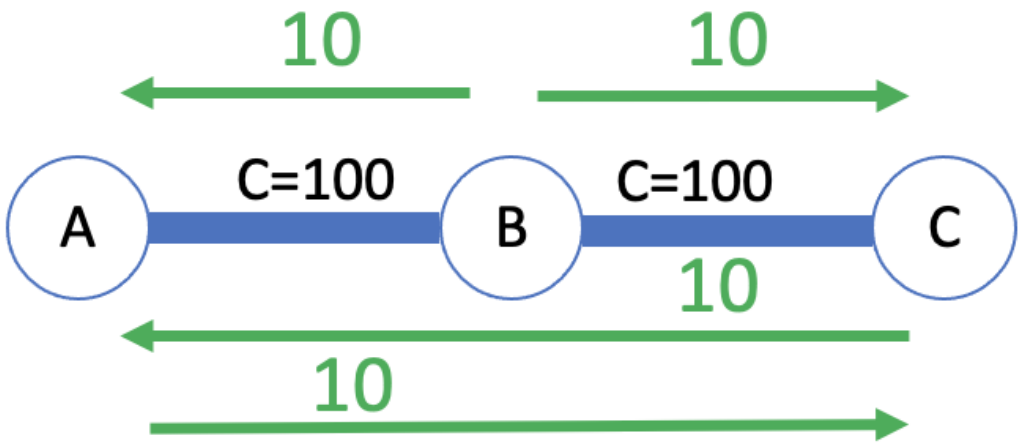


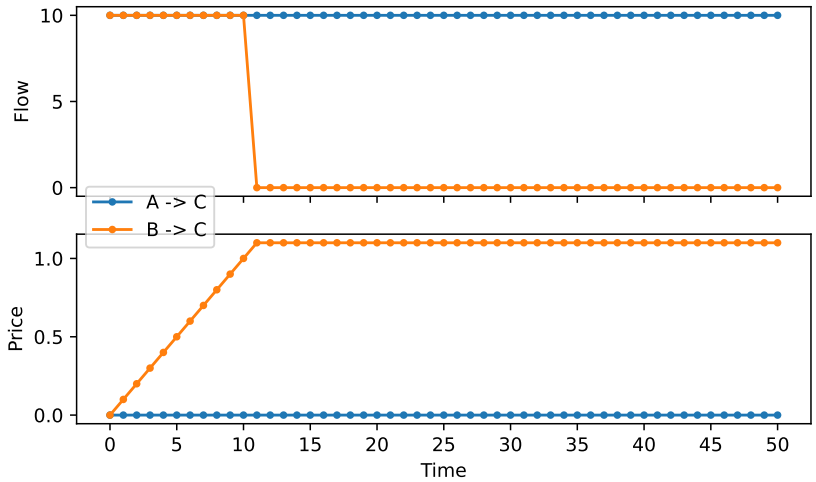


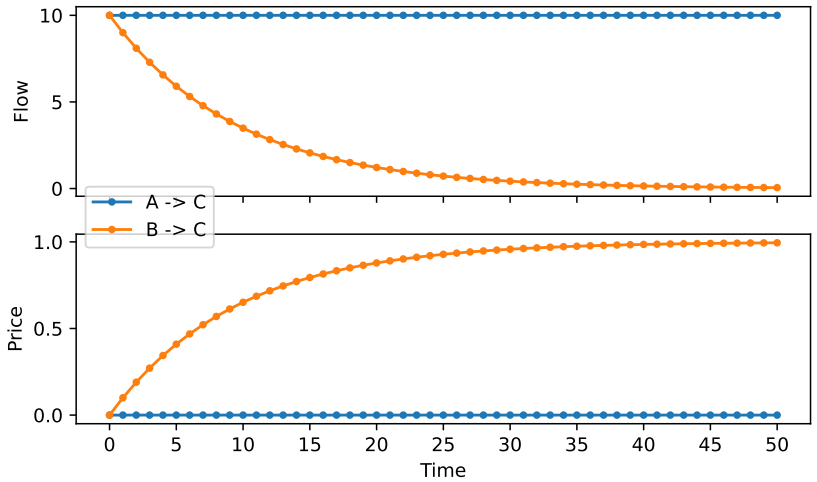
Simulations for two link deadlock example



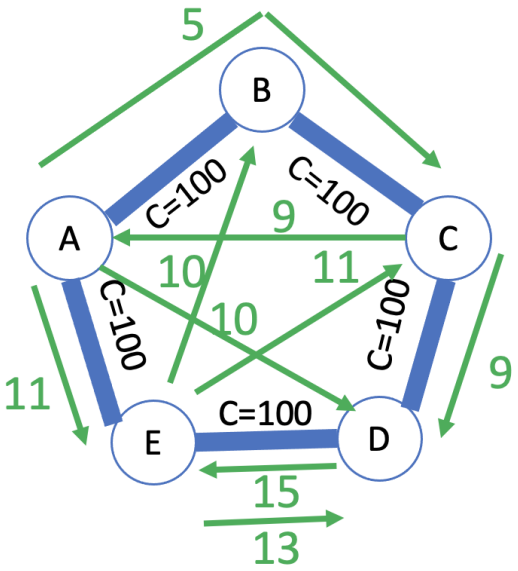
Simulations for two link deadlock example



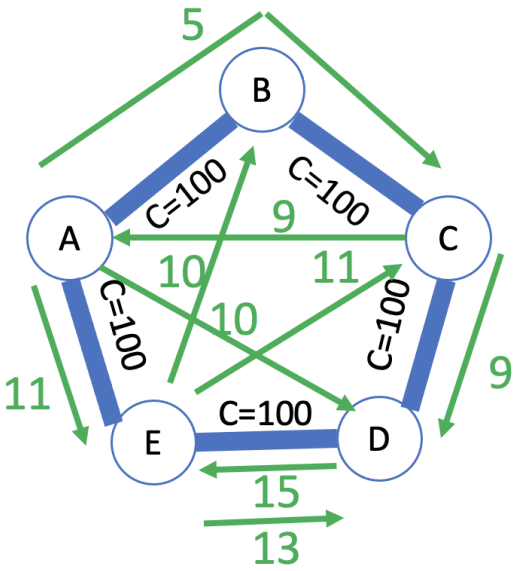


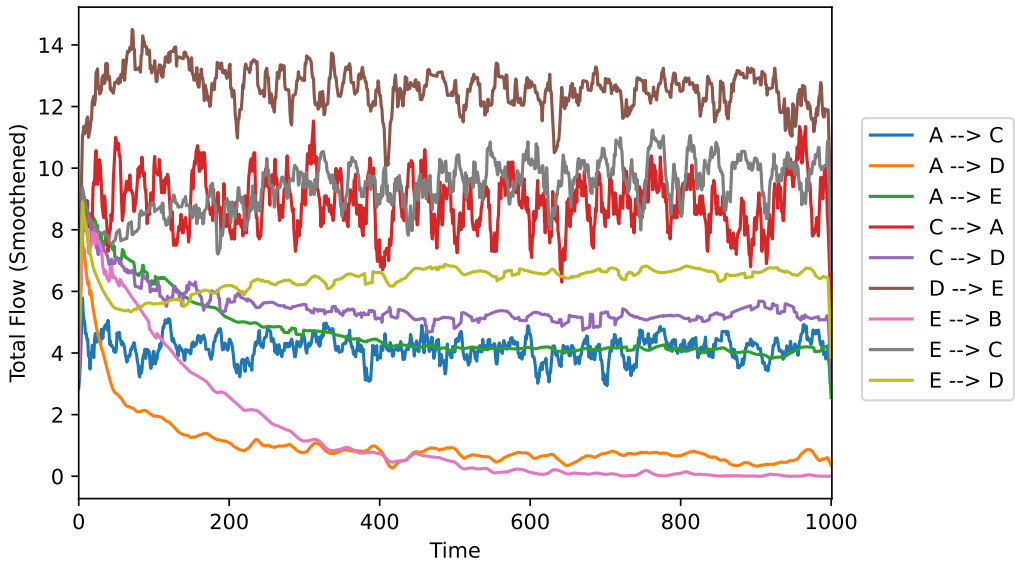


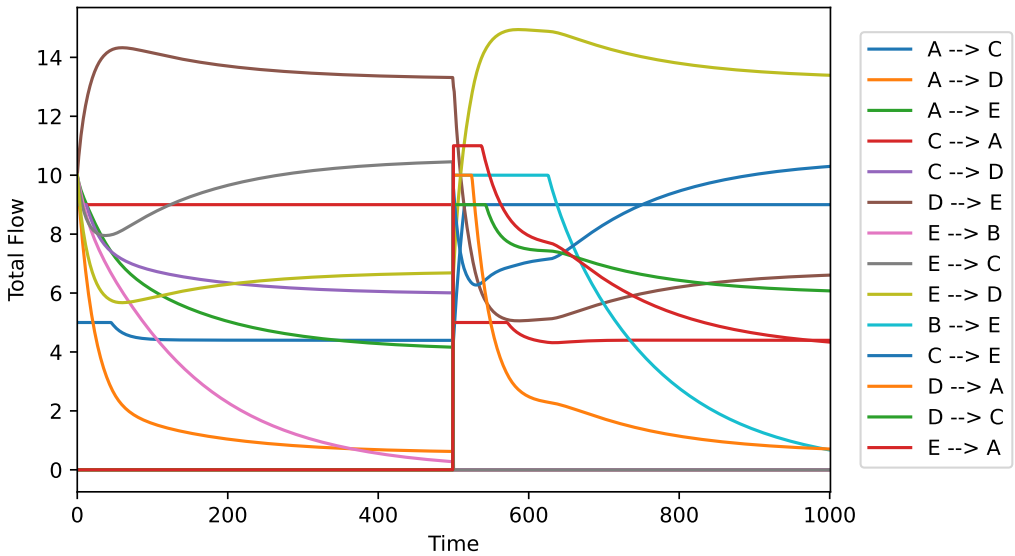
Simulations for five node network



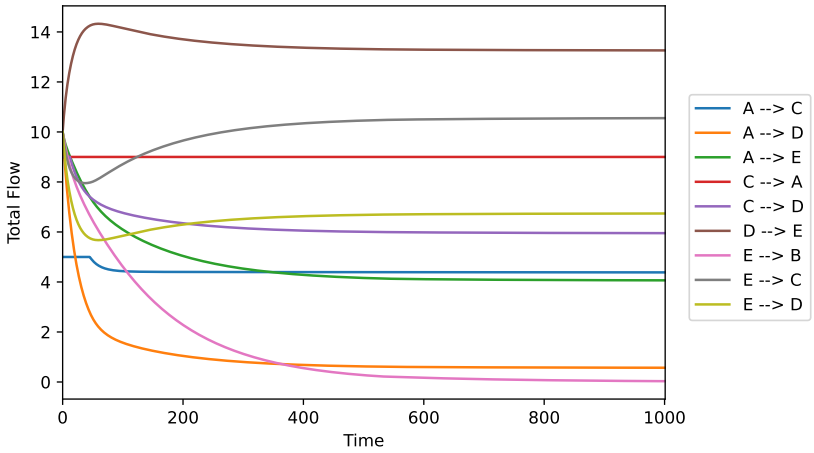
Simulations for five node network

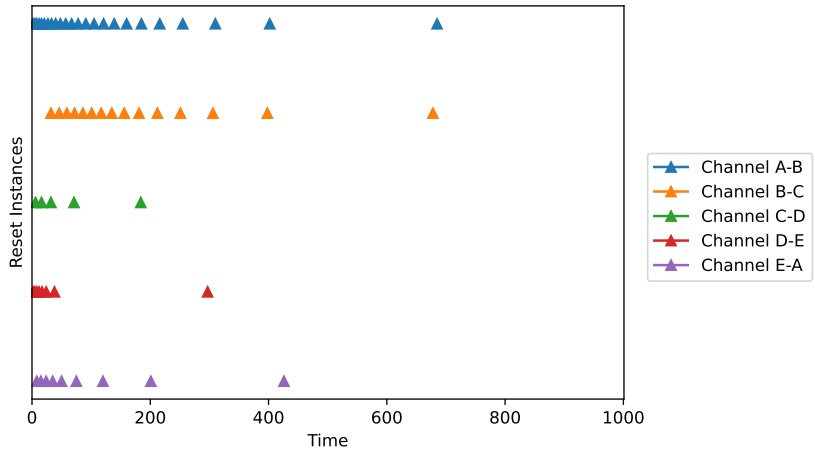


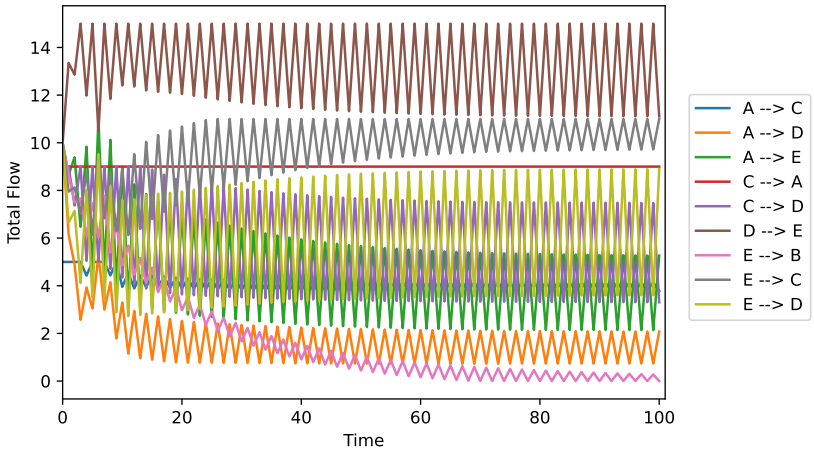


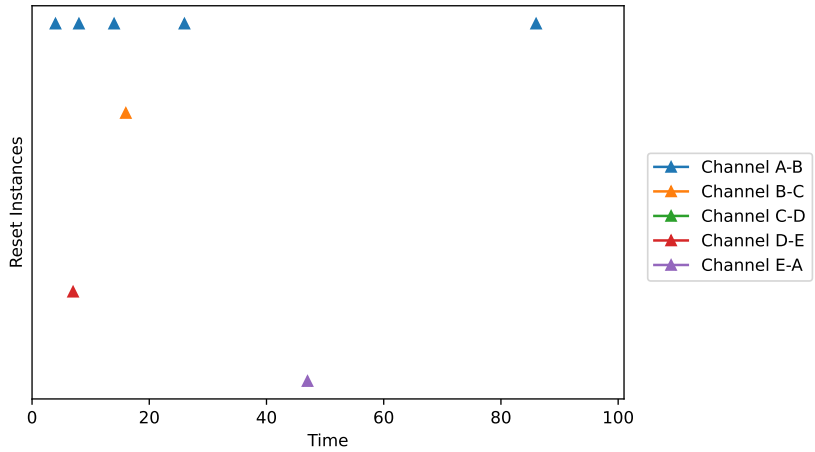


Simulations for five node network (continued)

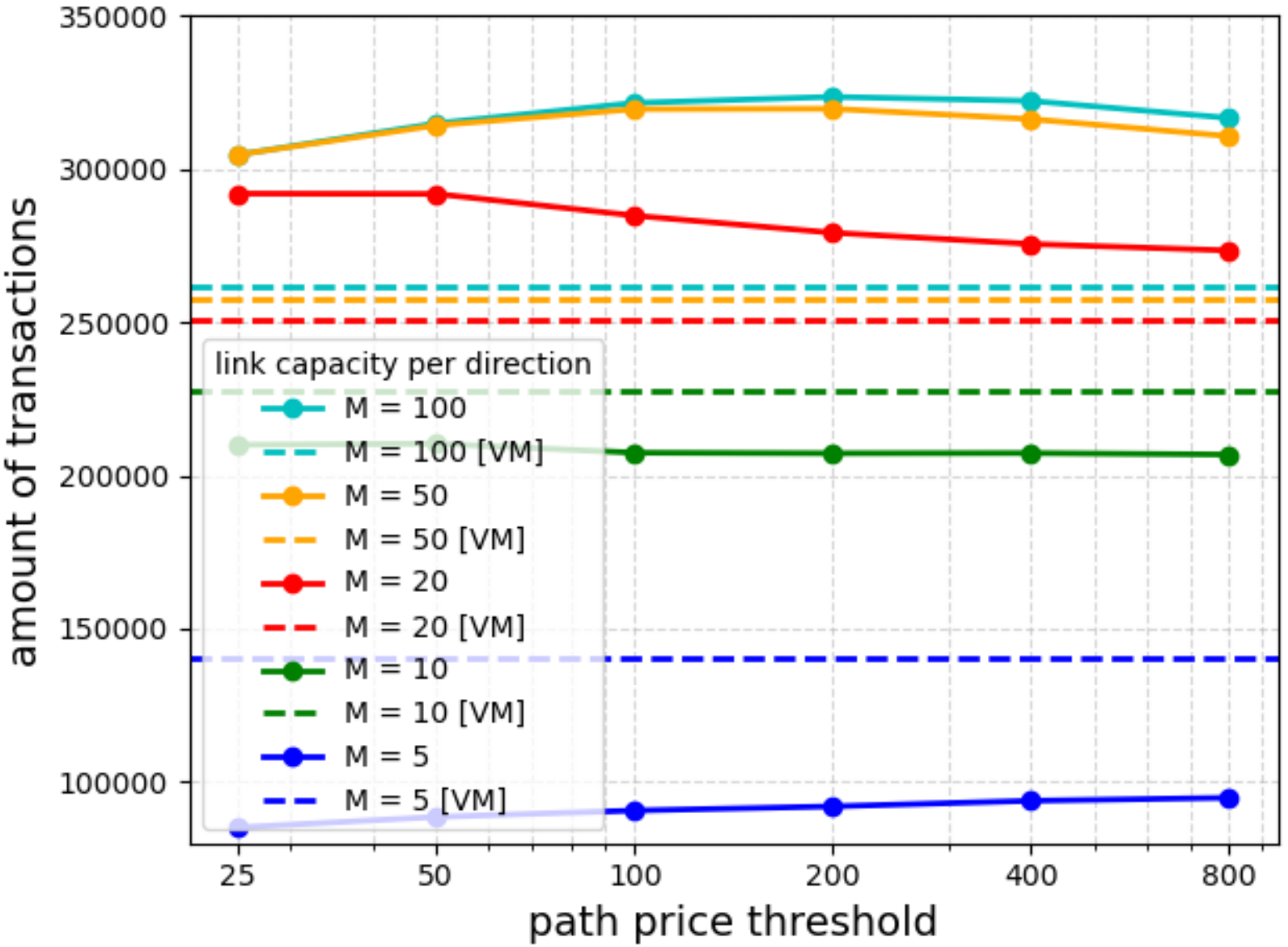








Simulations for ten node network with 1/3 acyclic demand



Conclusion

- A payment channel network cannot support arbitrary demands indefinitely without persistent channel resets
- The DEBT control protocol is joint routing and flow-control protocol that guides the network to a sustainable, stationary, optimal flow
- Protocol assumes users respond rationally to channel prices
- If a channel is rebalanced DEBT does not adjust channel price – price is proportional to long term imbalance of flow through channel
- Revising price update rules keeping privacy concerns, deadlock avoidance, and channel incentives are future avenues for research

THANK YOU!