Algebraic-Topological Formulation and Distributed Control of Packet-Switched Networks of General Topology

Dr. Yi-Ju Chao and Dr. Blaise Morton

yiju@mortoncorp.com; blaise@mortoncorp.com

Morton Consulting Corporation, Minnetonka, MN 55345

Tel: (952) 238-0800

Presented at MCIM and IMA Industrial Problem Seminar,

21 Jan 2005, University of Minnesota

Outline

- Packet-Switched Network Overview
- Challenges to model and control packet-switched networks
- Formulations
  - Multi-commodity network flow problems (prior art)
  - Queuing theory (prior art)
  - Algebraic Topological Formulation
- A Distributed Control System
  - Open-loop cycle control
  - Inner-loop coboundary control
- Conclusion / Future Directions

Packet-Switched Network Overview

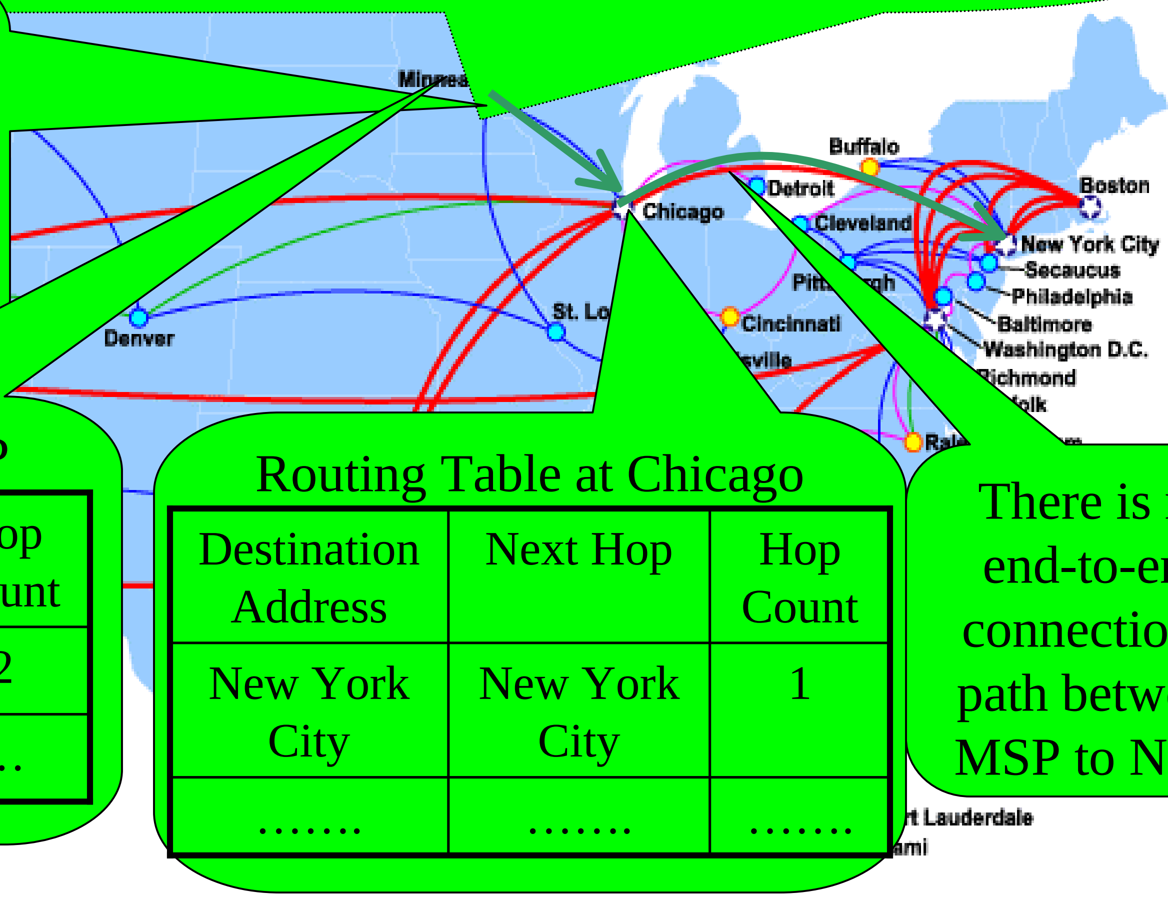
Courtesy Allegiance Telecom

Packet-Switched Network

Address: 220.151.102.25 – 11011100.10010111.01100110.00011001 to New York City

Packet	▪▪▪▪▪	Source IP address	Destination IP address	▪▪▪▪▪

Courtesy Cisco Systems, Inc.



Routing Table at MSP

Destination Address	Next Hop	Hop Count
NYC	Chicago	2
…….	…..	…

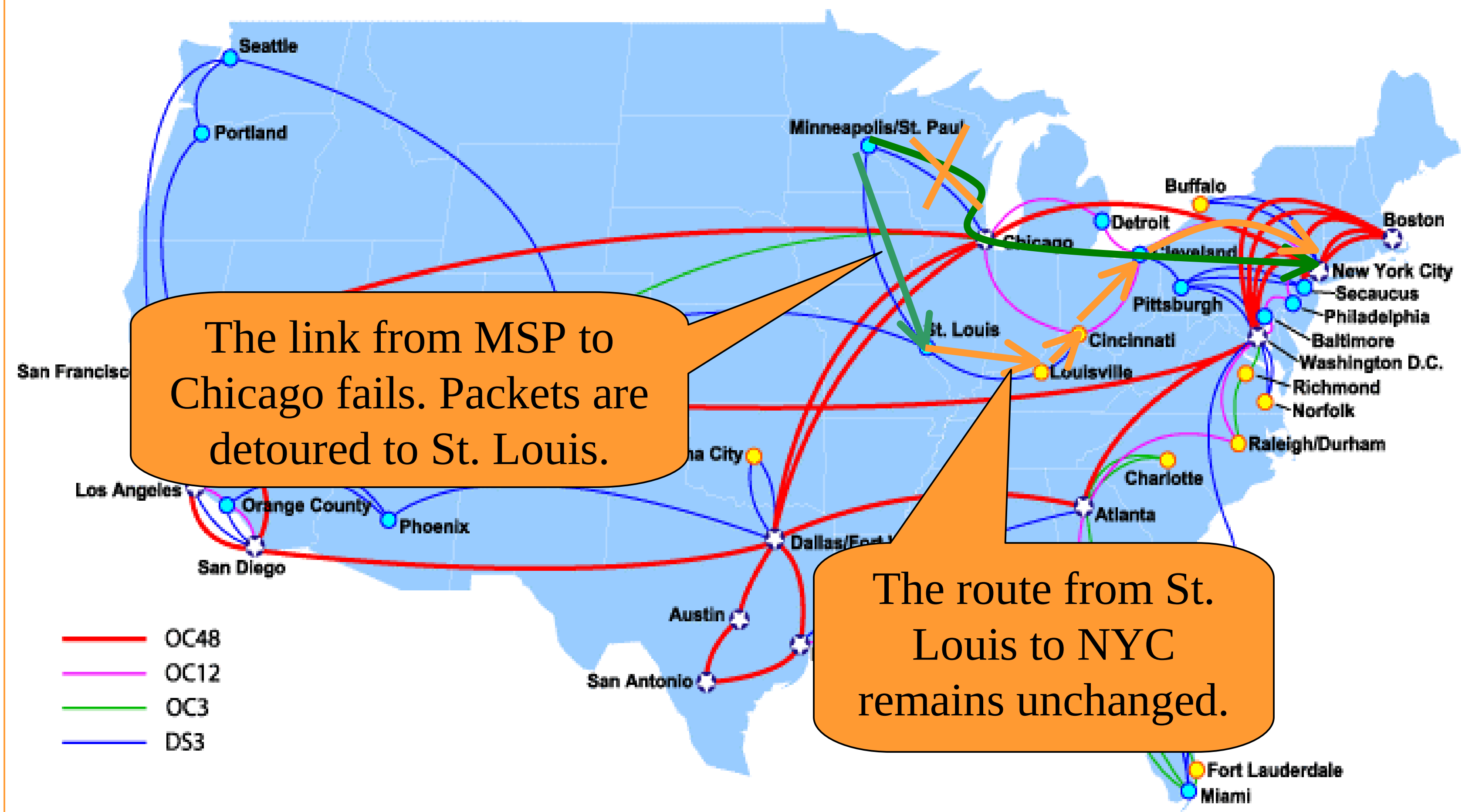
Routing Table at Chicago

Destination Address	Next Hop	Hop Count
New York City	New York City	1
…….	…….	…….

There is no end-to-end connectional path between MSP to NYC

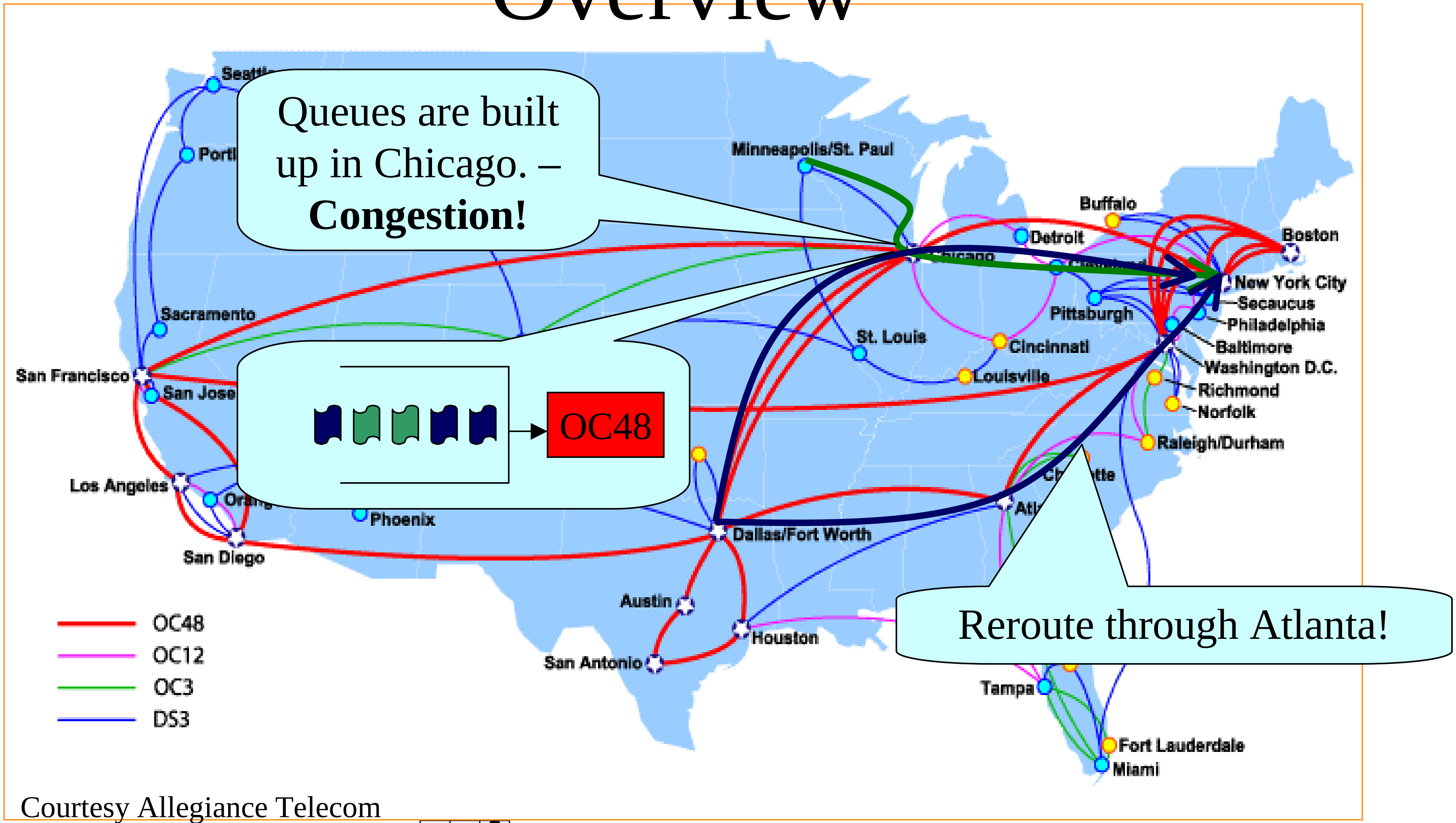
Courtesy Allegiance Telecom

Packet-Switched Network Overview



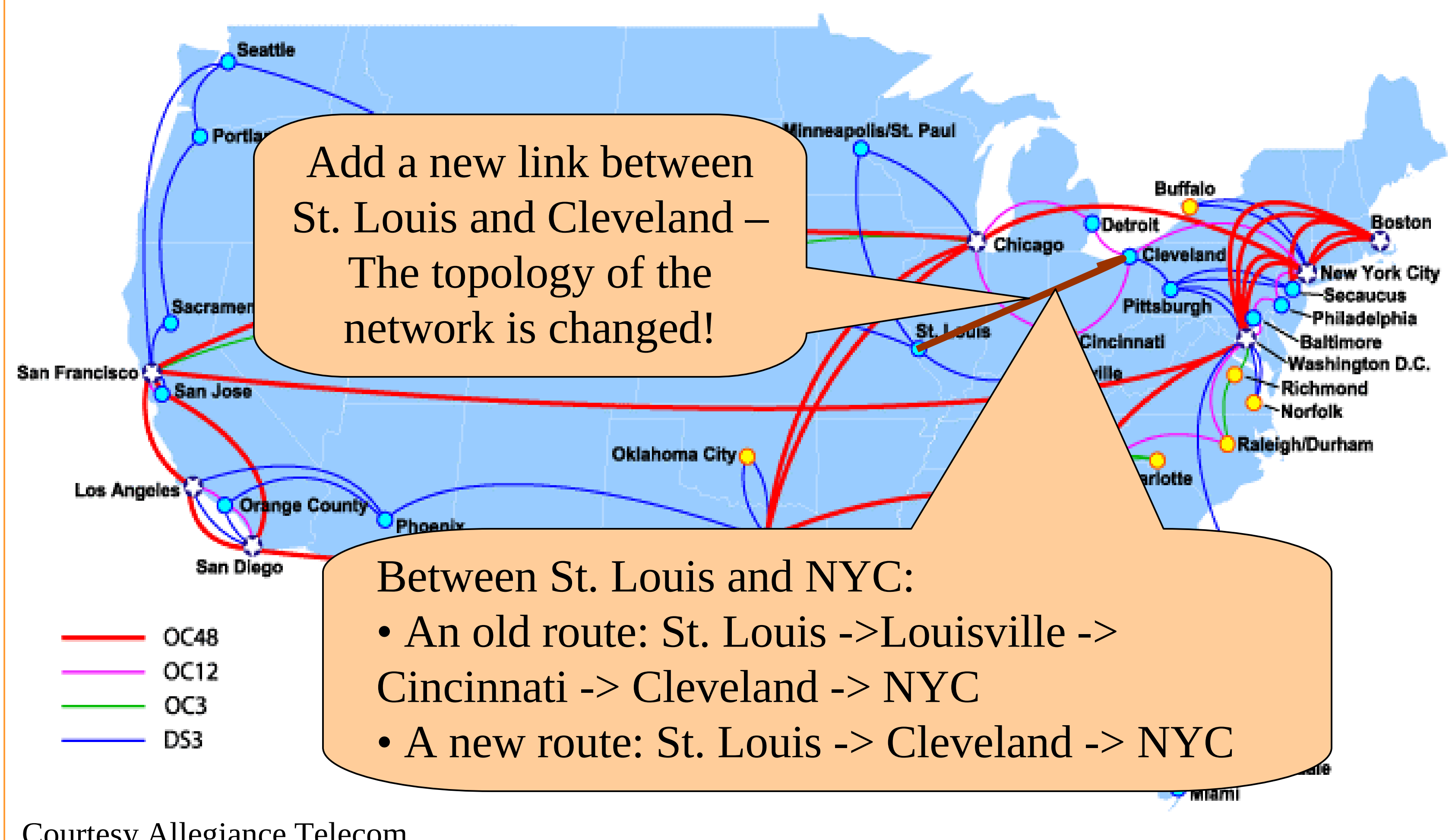
Courtesy Allegiance Telecom

Packet-Switched Network Overview



Courtesy Allegiance Telecom

Packet-Switched Network Overview



Add a new link between St. Louis and Cleveland – The topology of the network is changed!

Between St. Louis and NYC:

- An old route: St. Louis ->Louisville -> Cincinnati -> Cleveland -> NYC
- A new route: St. Louis -> Cleveland -> NYC

Courtesy Allegiance Telecom

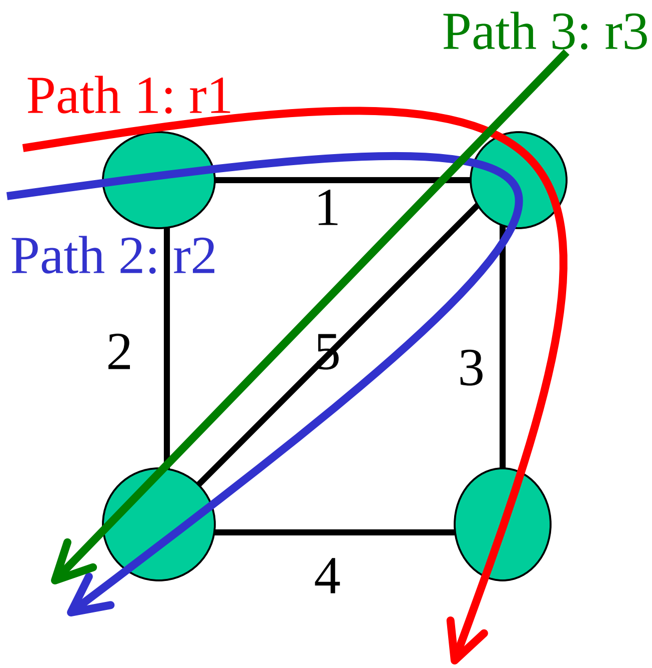
Routing Table Configuration

- Routers are geographically spread out
- Network topology changes
  - Link failure
  - Addition of new links
- Network congestion states are dynamic
  - Different bandwidth demands at different locations and different time
- Need to configure the routing table of each router locally, autonomously
  - Distributed or decentralized computation
  - Information for computation communicated between routers should be minimized

Challenges

- How to systematically incorporate **distributed** controls
  - (Static/Dynamic) route configuration
  - Admission control
  - Flow control
  - Congestion control
- Need a good methodology to analyze the effect of local controls on bandwidth utilization and overall network performance, which is a global problem
- Can a control system fulfill requirements (delay upperbound, minimal packet drop rate, bandwidth utilization, etc.)?

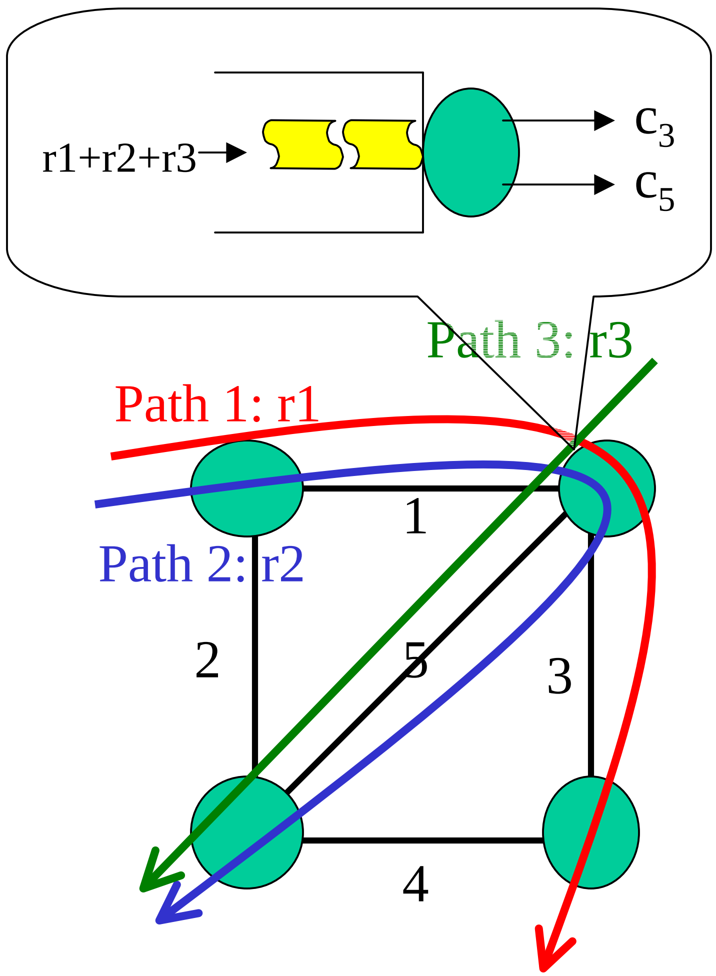
Multi-Commodity Network Flow Problem (Prior Art)



- This is **not** a distributed formulation for packet-switched networks. The model has predetermined end-to-end paths.
- In reality, R is **not** likely known; especially, R changes with topology changes and adaptive route configuration to traffic demands.
- Queues are **not** modeled.

$$\min_{y_i} \sum_{i=1}^{5} D_i(y_i, c_i), \quad y = Rr = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}$$

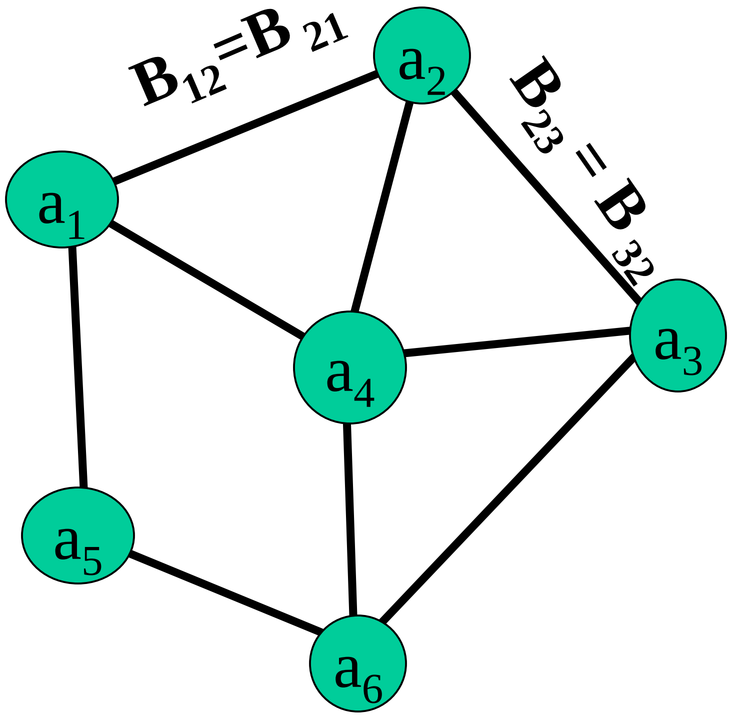
Queuing Theory (Prior Art)



- Usually assume probabilistic independence **<u>everywhere</u>**
  - Independence between nodes
  - Arrival rates and service rates are independent
- **<u>Hard</u>** to justify the connection between link capacities and service rates
- Most analysis relies on essential Markovian assumption, which is **<u>not</u>** observed for packet-switched networks

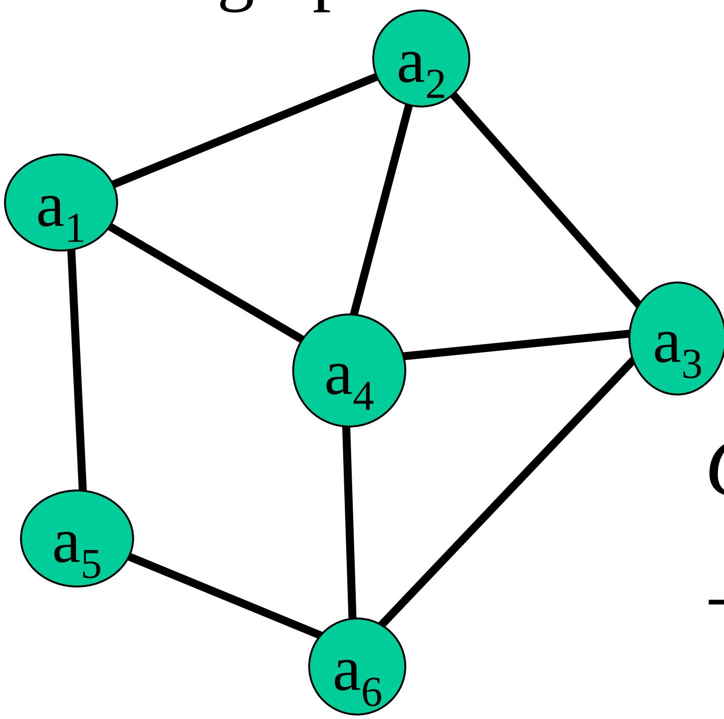
Algebraic Topological Notation

Consider a (symmetric) weighted graph G



0-chain: Node Space

G: a graph

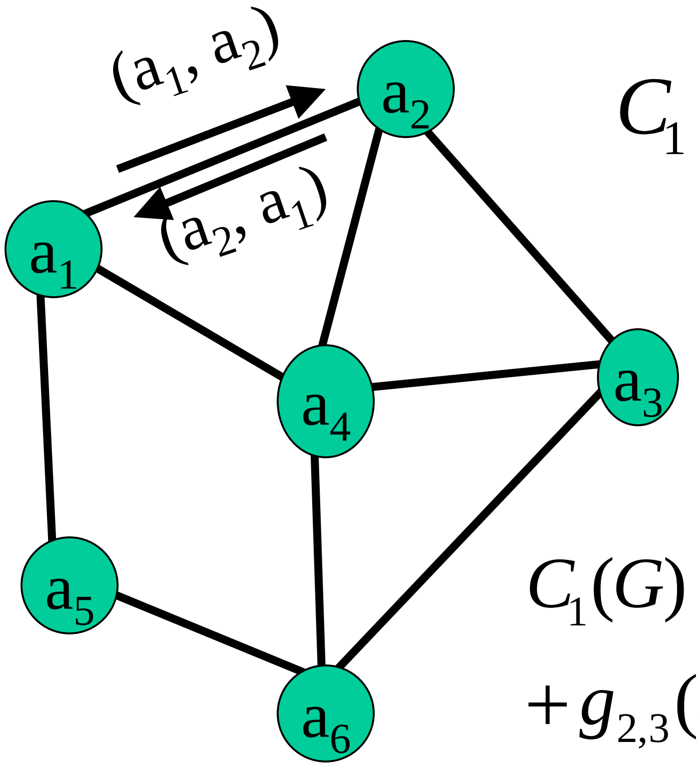


$$C_0 := \{ \sum_{i=1}^{N} f_i a_i \mid f_i \in \Re \}$$

Dimension = N = Number of nodes

$$C_0(G) = f_1 a_1 + f_2 a_2 + f_3 a_3 + f_4 a_4 + f_5 a_5 + f_6 a_6$$

1-chain: Link Space



$$C_1 := \{\sum_{n<m}^{M} g_{n,m}(a_n, a_m) \mid g_{n,m} \in \Re\}$$

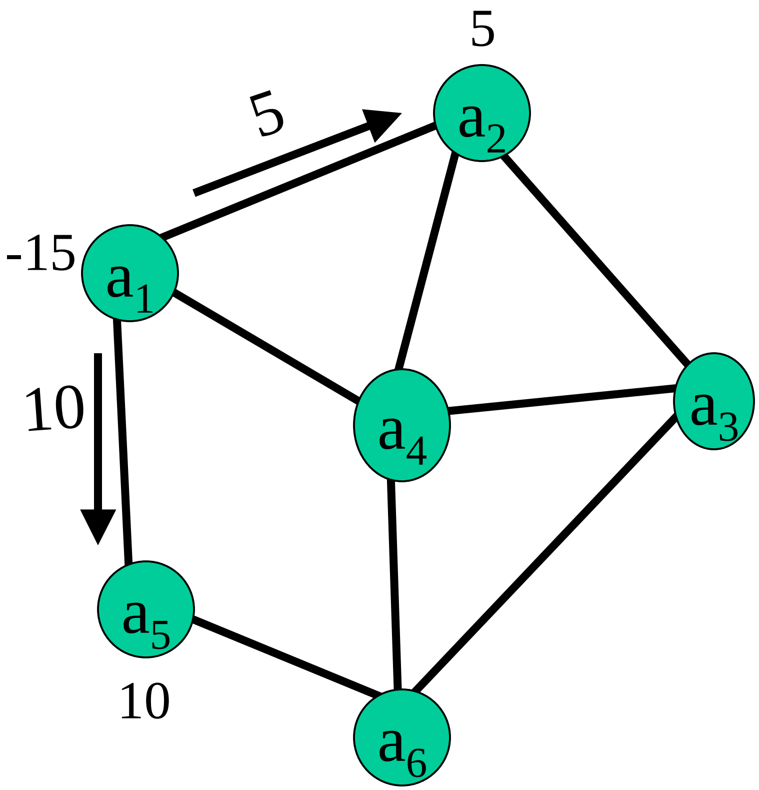
where $(a_i, a_j) = -(a_j, a_i)$

Dimension = M = Number of links

$$C_1(G) = g_{1,2}(a_1, a_2) + g_{1,4}(a_1, a_4) + g_{1,5}(a_1, a_5)$$
$$+ g_{2,3}(a_2, a_3) + g_{2,4}(a_2, a_4) + g_{3,4}(a_3, a_4)$$
$$+ g_{3,6}(a_3, a_6) + g_{4,6}(a_4, a_6) + g_{5,6}(a_5, a_6)$$

Boundary Operator: ∂

$$\partial_B : C_1 \to C_0, \partial_B(\sum x_{ij}(a_i, a_j)) = \sum x_{ij} B_{ij}(a_j - a_i)$$

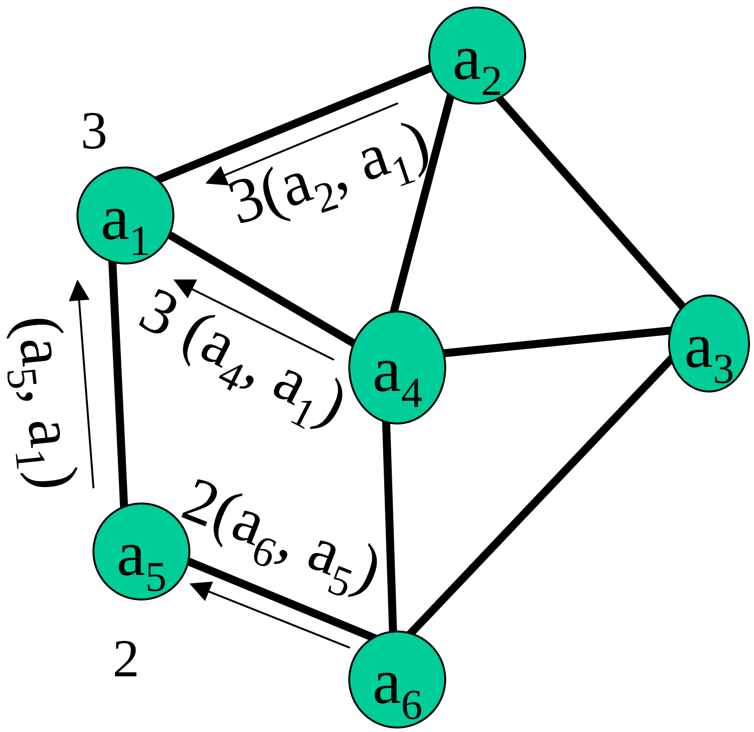


- ∂_B is a **local** linear operator, for example, $B_{ij} = 1$,

$$\partial_B(5(a_1, a_2) + 10(a_1, a_5))$$
$$= 5a_2 - 5a_1 + 10a_5 - 10a_1$$
$$= 5a_2 + 10a_5 - 15a_1$$

Coboundary Operator: δ

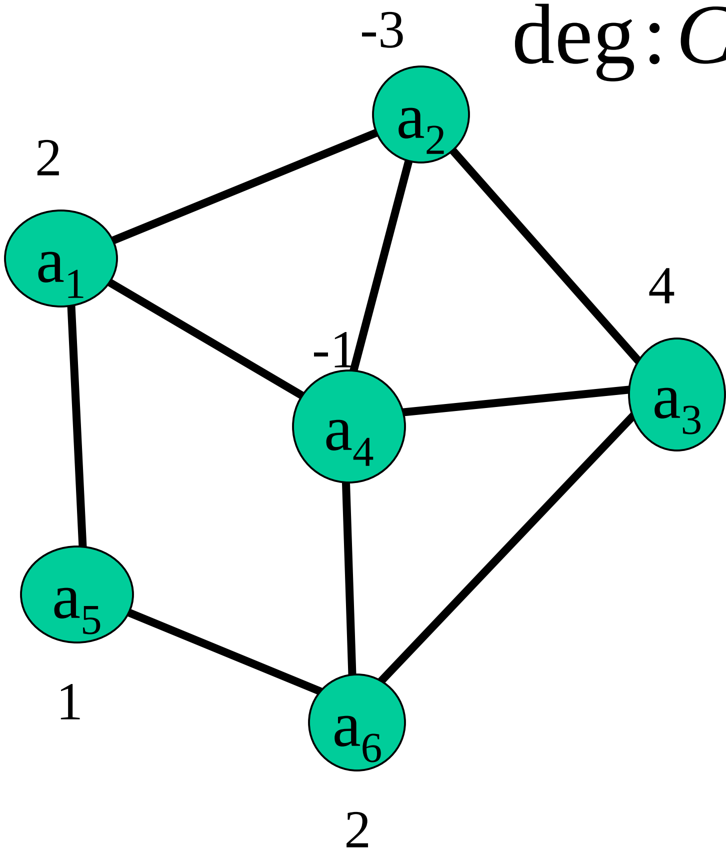
$$\delta_B : C_0 \rightarrow C_1, \delta_B(\sum_i f_i a_i) = \sum_i \sum_{a_j \in N(a_i)} f_i B_{ij}(a_j, a_i)$$



- δ_B is a **local** linear operator, for example, $B_{ij} = 1$,

$$\delta_B(3a_1 + 2a_5)$$
$$= 3(a_2, a_1) + 3(a_4, a_1) + 3(a_5, a_1)$$
$$+ 2(a_1, a_5) + 2(a_6, a_5)$$
$$= 3(a_2, a_1) + 3(a_4, a_1)$$
$$- 1(a_1, a_5) + 2(a_6, a_5)$$

Degree Operator: deg



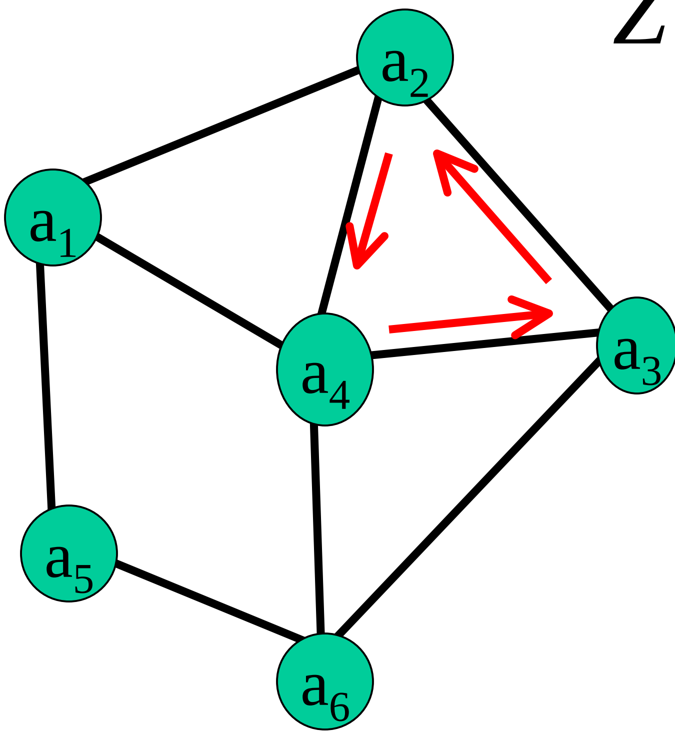
$$\deg : C_0 \rightarrow \Re, \deg(\sum_i y_i a_i) = \sum_i y_i$$

is a **global** linear operator

$$\deg(2a_1 - 3a_2 + 4a_3 - a_4 + a_5 + 2a_6)$$
$$= 2 - 3 + 4 - 1 + 1 + 2$$
$$= 5$$

- The dimension of the space ker(deg) is N -1, where N is the number of nodes.

Cycle Space: Z



$$Z_B(G) = \ker \partial_B \in C_1$$

- For example, $B_{ij} = 1$,

$5(a_2, a_4) + 5(a_4, a_3) + 5(a_3, a_2)$

is a cycle.

- The dimension of the cycle space of this graph is 4 = **M-N +1**, where N = 6 (number of nodes) and M = 9 (number of links)

Network Model Notation

- For simplicity of notation, here we assume all packets are destined for the same target node and are of the same quality-of-service class (everything does generalize).
- Consider a discrete-time network model, at time step *s* we use the following notation: at time s,
- $X_s \in C_0$ is the number of packets at nodes including queue (Q_s) and packets in process $(X_s - Q_s)$.
- $U_s \in C_0$ is the exogenous input rate vector, number of exogenous input packets at nodes.
- $Y_s \in C_0$ is the exogenous output rate vector, number of exogenous output packets at nodes.
- $T_s \in C_1$ is the transport rate vector, packets routed over links (using routing table stored at each node).

Network Balance Equations

- For each class of packets, for any routing policy T, the **Fundamental Dynamic Balance Equation** (law of conservation) is true in deterministic models or pathwise in stochastic models,

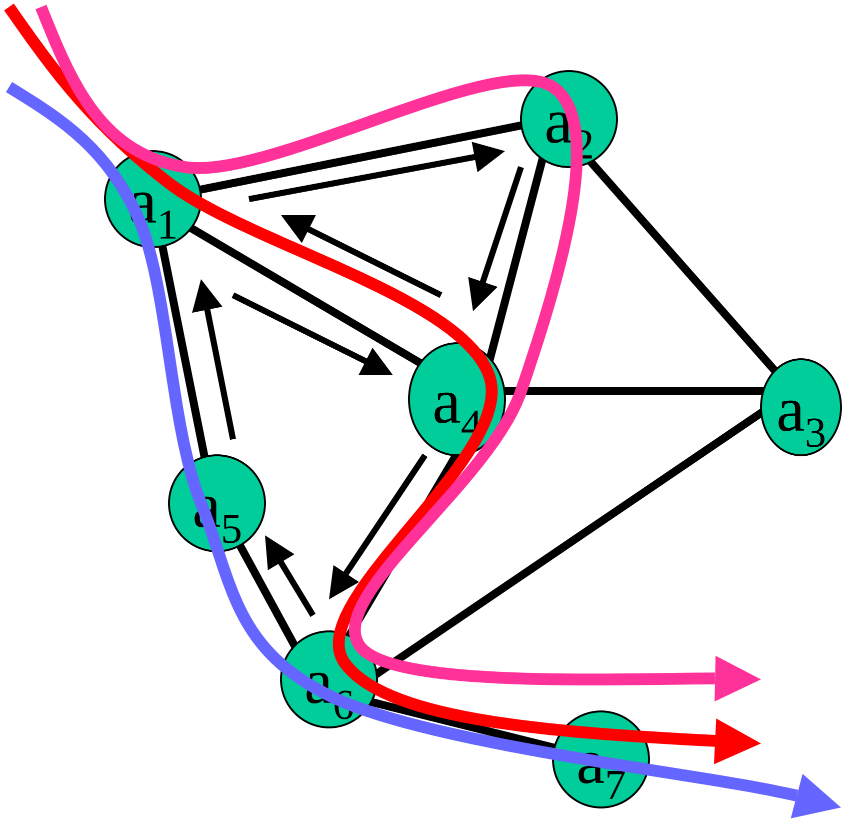
$$\frac{dX}{dt} = U - Y + \partial(T) \quad \text{where } \partial = \partial_I$$

- Network **Steady-State Balance Equation**

$$U - Y + \partial(T) = 0,$$

$$Y = \deg(U)$$

A Simple Observation

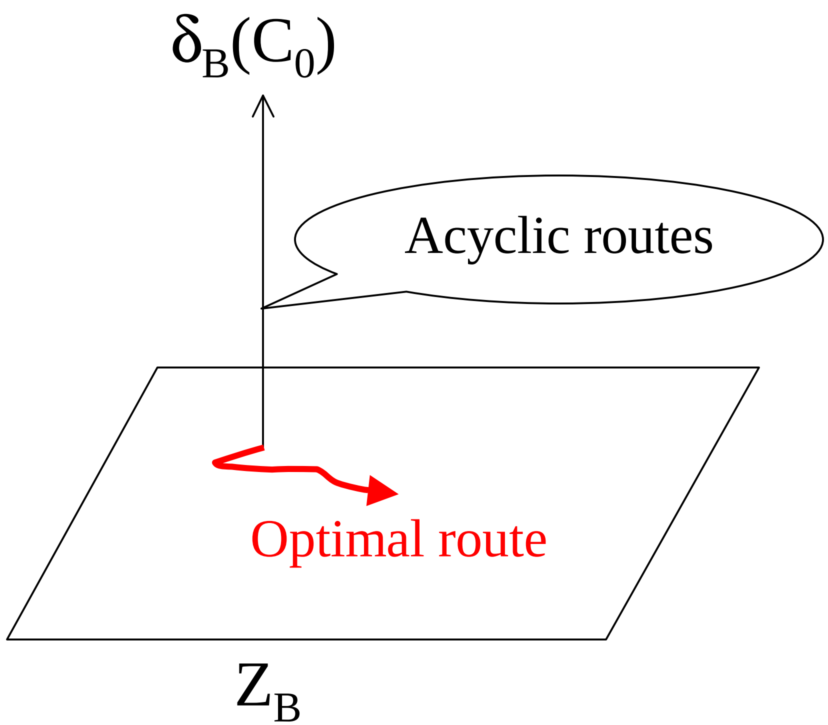


- For a fixed network flow, pick an arbitrary route T for the flow. An optimal route can be achieved by adding cycles on the route.
- The cycle space is of dimension M – N + 1 .

Route Decomposition

$$C_1 = \delta_B(C_0) \oplus \ker(\partial_B) = \delta_B(C_0) \quad \oplus \quad Z_B$$

dim=N-1   dim=M-N+1



- An optimal routing solution T w.r.t. any cost function can be obtained by adding cycles to a route in $\delta_B(C_0)$
- Any route in $\delta_B(C_0)$ is acyclic
- The space of acyclic routes $\supset \delta_B(C_0)$

A Distributed Control System

- Outer-loop **Cycle** control: low-bandwidth outer-loop provides good steady-state performance
  - A distributed algorithm generates optimal steady-state routes
- Inner-loop **Coboundary** control: high-bandwidth inner-loop improves network stability
  - A distributed algorithm generates route adjustments using queuing as feedback in order to relax network congestion
- Network stability – Balance Equations should be satisfied
- Acyclic flows – Every flow lying in $\delta_B(C_0)$ is acyclic
- How to incorporate admission control and flow control in the above control system?

Comparison of Optimization Problems

Optimization over path space

$$\min_{(x_p)} \sum_{(i,j)} D_{ij}[\sum_{\substack{all\ paths\ p \\ containing\ (i,j)}} T_p\ ]$$

$$subject\ to \sum_{p\in P_w} T_p = r_w, \forall w$$

$$T_p \geq 0, \forall p \in P_w, w \in W$$

• W: the set of all Source-Destination (SD) pair nodes

• P_w: the set of all directed paths connecting the SD pair $w \in W$

• T_p: the flow of the path p (bits/sec)

• r_w: the arrival rate of the SD pair w (bits/sec)

Optimization over link space

$$\min_{(x^k)} \sum_{(i,j)} D_{ij}[T_{ij}^1, \ldots, T_{ij}^K],\ \ T^k = \sum_{(i,j)} T_{ij}^k(a(i), a(j))$$

subject to

$$\partial(T^k) + U^k - \deg(U^k)a(N_k) = 0$$

$$\forall k = 1, \ldots, K, \forall i, j, k$$

• U^k: the arrival rate of k-class packets at nodes, destined to the node $a(N_k)$ (bits/sec)

• T^k : the flows of k-class packets at links (bits/sec)

• T^k_{ij} : the flows of k-class packets at link (i, j) (bits/sec)

Optimal Solutions over Link Space

- In case of linear cost function, Bellman-Ford or Dijkstra's distributed algorithms provide solutions.
- In case of quadratic cost function $\sum_{(i,\, j)}(T_{ij}/B_{ij})^2$, the optimal routing policy is

$$T = B.*\ \delta_B A_B(\deg(U)\ a(N) - U),$$

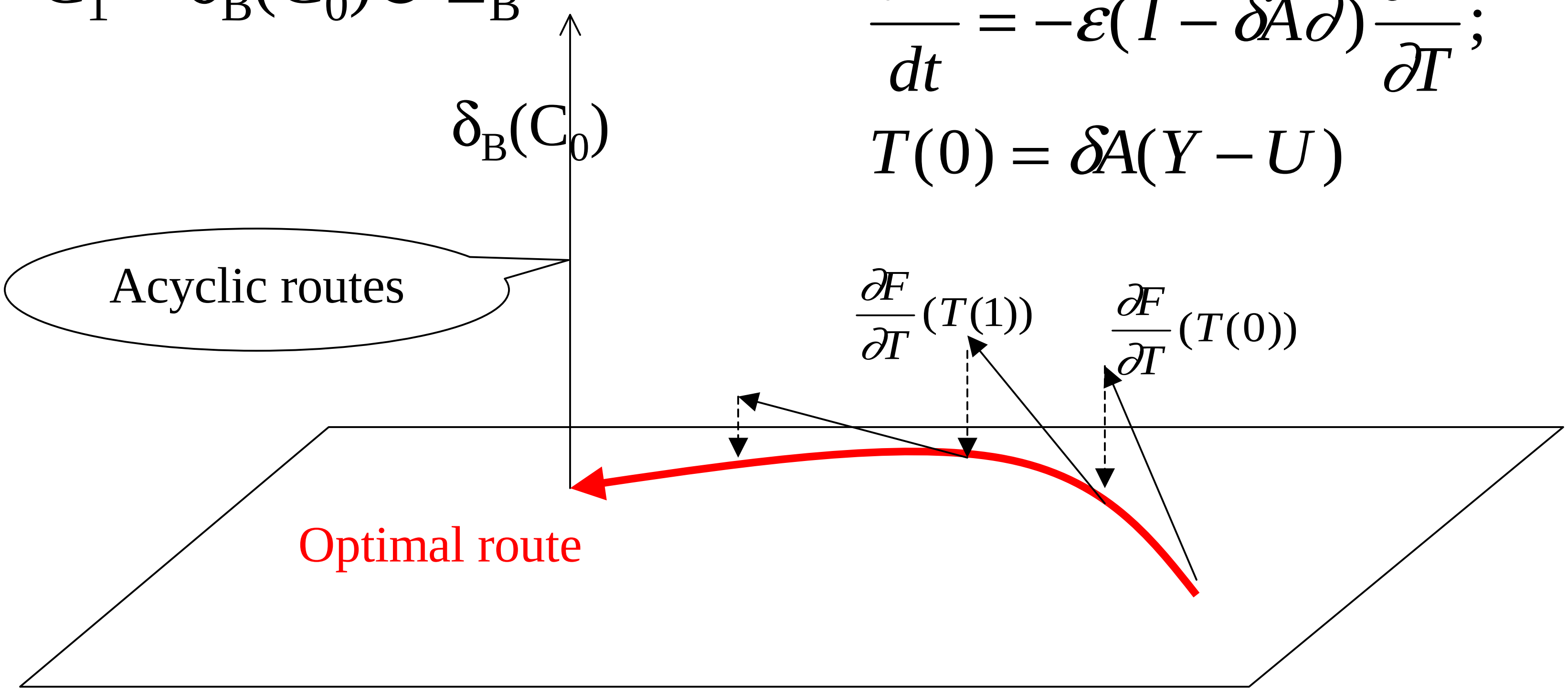
  where A_B (global operator) is the inverse of $\partial_B\, \delta_B = L_B$ (Laplacian) in ker(deg).
- Quadratic cost functions are applied to electric-circuit networks
- Solutions of well-posed general cost functions can be obtained by constructing a first-order linear system obtained by projecting the gradients of the cost function to the space of cycles.

Optimization of General-Cost Functions

$$C_1 = \delta_B(C_0) \oplus Z_B$$

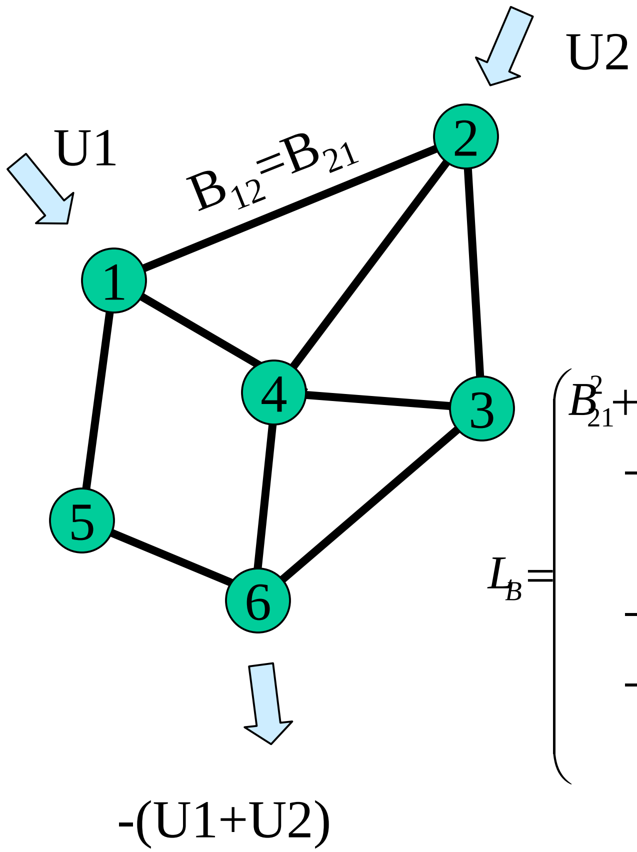
$$\frac{dT}{dt} = -\varepsilon(I - \delta A \partial)\frac{\partial F}{\partial T};$$

$$T(0) = \delta A(Y - U)$$



$$\{T: \partial T + U - Y = 0\} \approx Z_B$$

Examples of Laplacian



U = U1 a(1) + U2 a(2);

Y = - (U1+U2) a(6);

U – Y + ∂ T = U – deg(U) a(6) + ∂ T = 0

$$L_B=\begin{pmatrix} B_{21}^2+B_{41}^2+B_{51}^2 & -B_{21}^2 & 0 & -B_{41}^2 & -B_{51}^2 & 0 \\ -B_{12}^2 & B_{12}^2+B_{32}^2+B_{42}^2 & -B_{32}^2 & -B_{42}^2 & 0 & 0 \\ 0 & -B_{23}^2 & B_{23}^2+B_{43}^2+B_{63}^2 & -B_{43}^2 & 0 & -B_{63}^2 \\ -B_{14}^2 & -B_{24}^2 & -B_{34}^2 & B_{14}^2+B_{24}^2+B_{34}^2+B_{64}^2 & 0 & -B_{64}^2 \\ -B_{15}^2 & 0 & 0 & 0 & B_{15}^2+B_{65}^2 & -B_{65}^2 \\ 0 & 0 & -B_{36}^2 & -B_{46}^2 & -B_{56}^2 & B_{36}^2+B_{46}^2+B_{56}^2 \end{pmatrix}$$

L_B, called the “Laplacian”, is in fact $\partial_B \delta_B$, which is a local linear operator.

Main Result on Aperiodic Graphs

- Acyclicness and convergence of distributed algorithms to find optimal solutions rely on the following two theoretical results:
- **<u>Theorem (Aperiodic, not necessarily bidirectional graphs)</u>** Suppose (I - D^{-1} L) is aperiodic. For u $\in$ ker(deg) $\subset C_0$, let z = D^{-1} u, then

$$\varphi(z) = \lim_{k \to \infty} \sum_{j=1}^{k} \delta(1 - D^{-1}L)^j z + \delta z \in \mathrm{Im}(\delta) \subset C_1$$

is well defined and acyclic. Moreover, we have

$$\partial\varphi(z) = u$$

Main Results on Graphs of Period 2

- **Theorem (Bipartite graphs, periodic of period = 2)**

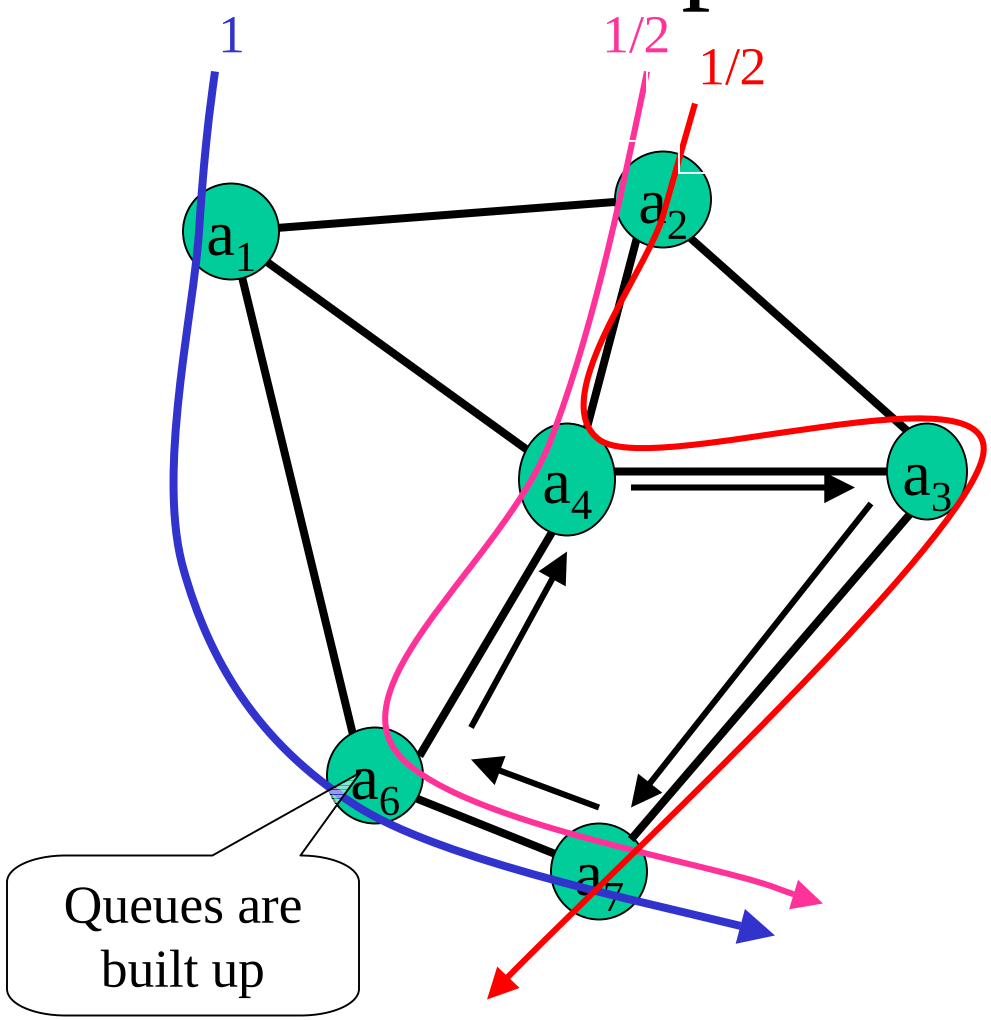
Suppose (I - D^{-1} L) is of period 2. For u $\in$ ker(deg) $\subset C_0$, let z = D^{-1} u, then

$$\varphi_1(z) = \lim_{k \to \infty} \sum_{j=1}^{2k} \delta(1 - D^{-1}L)^j z + \delta z \in \mathrm{Im}(\delta) \subset C_1$$

$$\varphi_2(z) = \lim_{k \to \infty} \sum_{j=1}^{2k+1} \delta(1 - D^{-1}L)^j z + \delta z \in \mathrm{Im}(\delta) \subset C_1$$

are well defined and acyclic. Let x = (φ_1(z) - φ_2(z))/2, then we have $\partial\varphi_1(z) + x = \partial\varphi_2(z) - x = u$

A Simple Observation

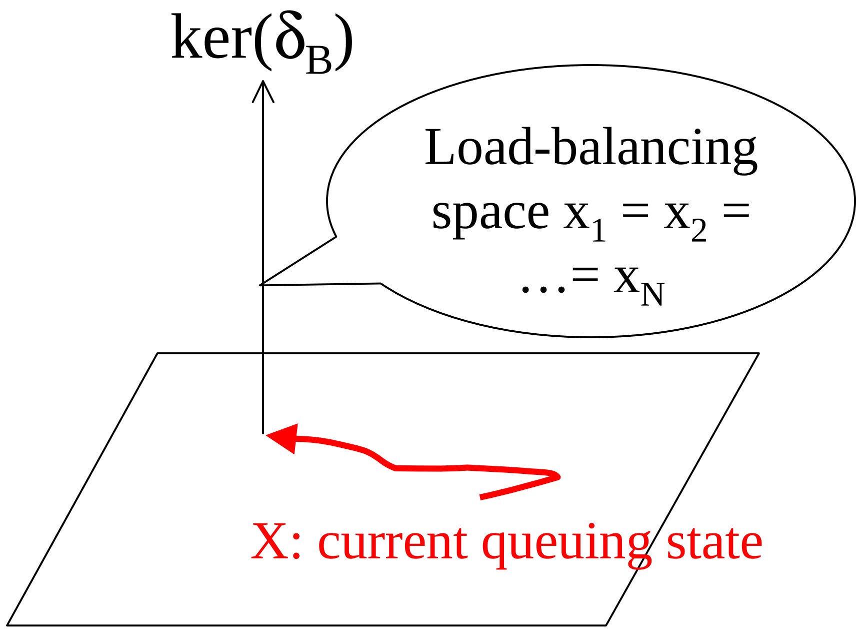


- Use dynamic routing to decongest queues
- If exogenous traffic inputs are too large, queues will build up and lead to instability.
- Multi-path routing (rather than switch routes) leads to more robust route changes

Load-Balancing Queues

$$C_0 = \partial_B(C_1) \oplus \ker(\delta_B) = \ker(\deg) \oplus \ker(\delta_B)$$

dim = N-1 dim = 1



$\ker(\deg) = \{x: \deg(x) = 0\}$

- The best we can do for any load-balancing scheme is to drive queues into $\ker(\delta_B)$

- Given a current queue state $X \in C_0$, the queue state can be moved to $\ker(\delta_B)$ by using feedback control, for some $\varepsilon > 0$,

$$\frac{dX}{dt} = -\varepsilon \partial_B \delta_B \left(\frac{X}{\overline{\overline{X}}}\right) = -\varepsilon L_B \left(\frac{X}{\overline{\overline{X}}}\right)$$

- The associated routing adjustment is

$$\Delta T = -\varepsilon B.\, \delta_B \left(\frac{X}{\overline{\overline{X}}}\right)$$

Summary of the Control System

Admission or flow control

Load-balancing queues

$$X = c\,\overline{X} + (X - c\,\overline{X}) \text{ ,where } c = \deg(X)/N$$

$$\mathbf{C_0 = \ker(\delta_B) \oplus \ker(\deg)}$$

$$N = 1 + (N-1)$$

Fundamental network dynamic:

$$\frac{dX}{dt} = U - Y + \partial(T)$$

Inner-loop (feedback)control dynamic:

$$\frac{dX}{dt} = -\varepsilon \partial_B \delta_B\left(\frac{X}{\overline{X}}\right) = -\varepsilon L\left(\frac{X}{\overline{X}}\right)$$

$$\Delta T = -\varepsilon B.\delta_B\left(\frac{X}{\overline{X}}\right)$$

$$M = (M-N+1) + (N-1)$$

$$\mathbf{C_1 = Z_B \oplus \delta_B(C_0)}$$

$$T = (Z_1 + T_1) + \Delta T$$

Outer-loop optimal control

Inner-loop route adjustments

Conclusions

- A new algebraic topological formulation for control of packet-switched networks is proposed.
- The fundamental dynamic equation for network flows is

$$\frac{dX}{dt} = U - Y + \partial(T)$$

- A distributed control system for packet-switched networks is proposed:
  - Outer-loop Cycle control: theoretical results support the convergence of iterative distributed algorithms for finding optimal solutions of algebraic topological network flow optimization
  - Inner-loop Coboundary control: using Laplacian to load balance queues by routing adjustments.
  - The link between the systematical control methodology and the decompositions the node space and the link is shown.

Future Directions (1)

- Network distributed control
  - We have performed simulation on a network of 100 nodes. Topology does matter in terms of convergence time and control robustness.
  - Robustness of the network control systems
    - Link failures, highly dynamic traffic environments
  - Detailed studies to link route configuration, congestion, flow and admission control.

Future Directions (2)

- Distributed computation
  - Incorporate stochastic inputs
  - Extension to cost functions with node cost as well
    - Can be solved also by establishing a first-order linear system using ∂, δA
  - Detailed studies on distributed algorithms to solve general cost functions
    - Asynchronous, well-posed conditions
  - The convergence rate of the distributed algorithm is related to the spectrum of $(I - D^{-1} L)$