Application-Specific 3D Network-on-Chip Design Using Simulated Allocation

Pingqiang Zhou, Sachin S. Sapatnekar
University of Minnesota

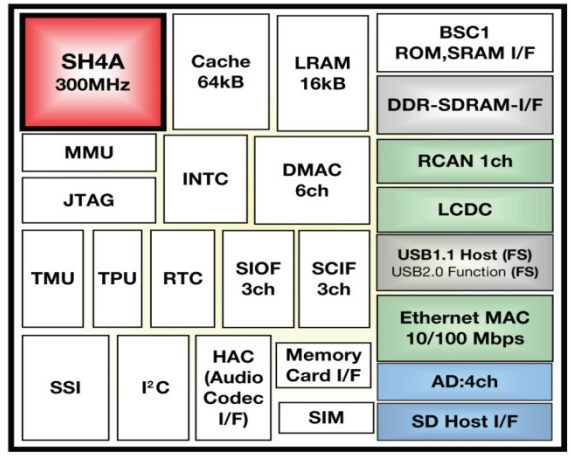
Ping-Hung Yuh
National Taiwan University

Communication Challenges in SoCs

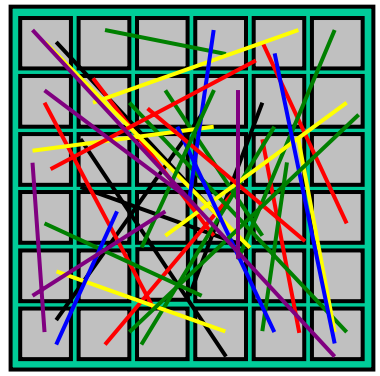
- ITRS predictions for SoCs
  - Composed of 10s-100s of cores
  - 50nm (or lower) technology
  - Clocks running at multi-GHz range

- Challenges to inter-core (global) communication by signal wires
  - Long signal propagation delays
    - Synchronous communication infeasible
  - Data errors due to signal integrity
    - Crosstalk, electromagnetic interference

- Bus based architectures undesirable
  - Synchronous
  - Not scalable



Euclid SH7397 SoC diagram

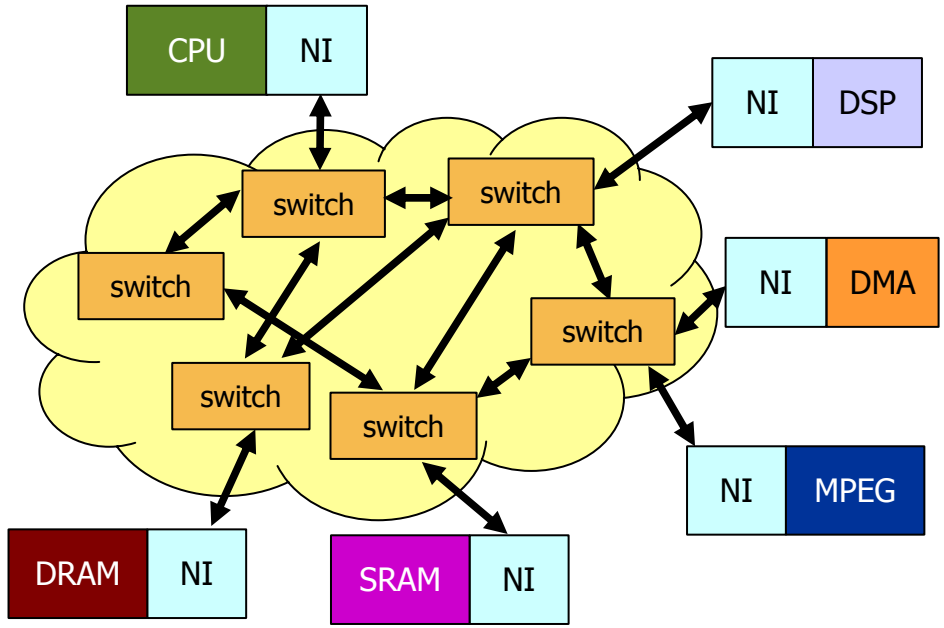


[Kolodny, 2005]

Interconnection by dedicated wires

Network-on-Chip (NoC)

- Solution to global communication challenges
- Packet-based, asynchronous communication
- Interconnect structure and wiring complexity can be well controlled
- Inherently scalable
- High bandwidth
  - Support concurrent communication



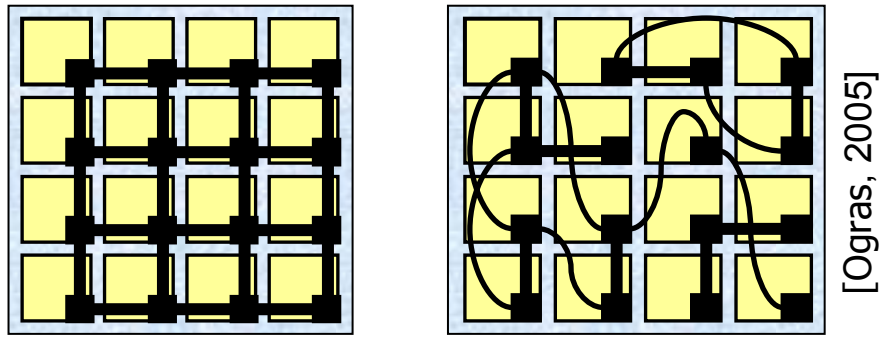
[Murali *et al.*, ICCAD, 2006]

NoCs needed for SoCs

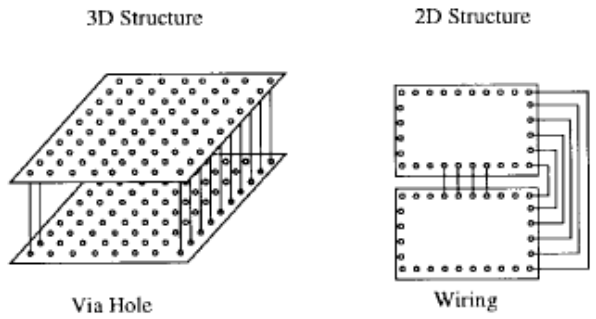
Adapted from [Srinivasan *et al.*, *ICCAD*, 2005]

NoC Architecture Design Issues

- Topology selection
  - Regular or custom?
- Optimize switches
  - Number, area, power etc.
- Route the traffic flows, deadlock-free
- Physical floorplanning of cores + switches
- 3D-specific challenges
  - Technology constraints, like TSV#
  - Tier assignment
  - Placement of switches
  - Accurate power and delay modeling

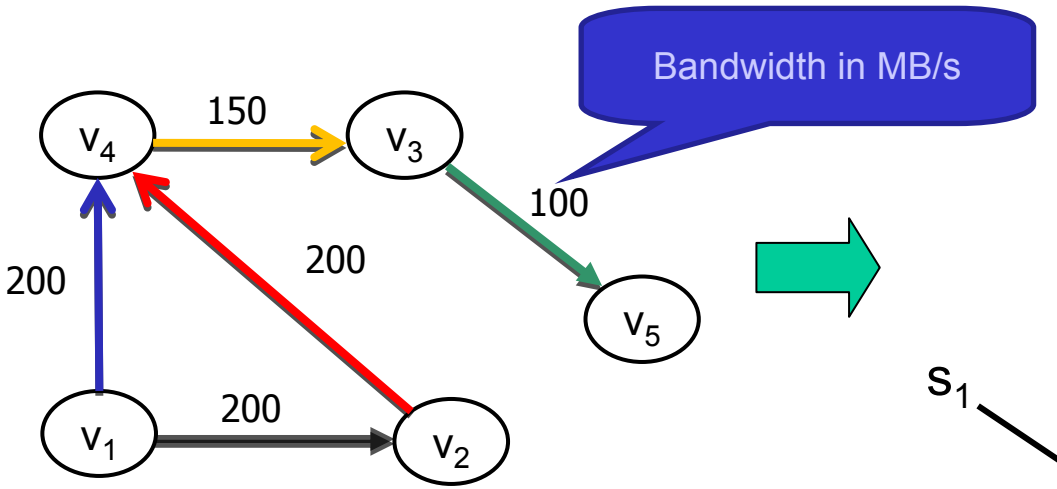


[Ogras, 2005]

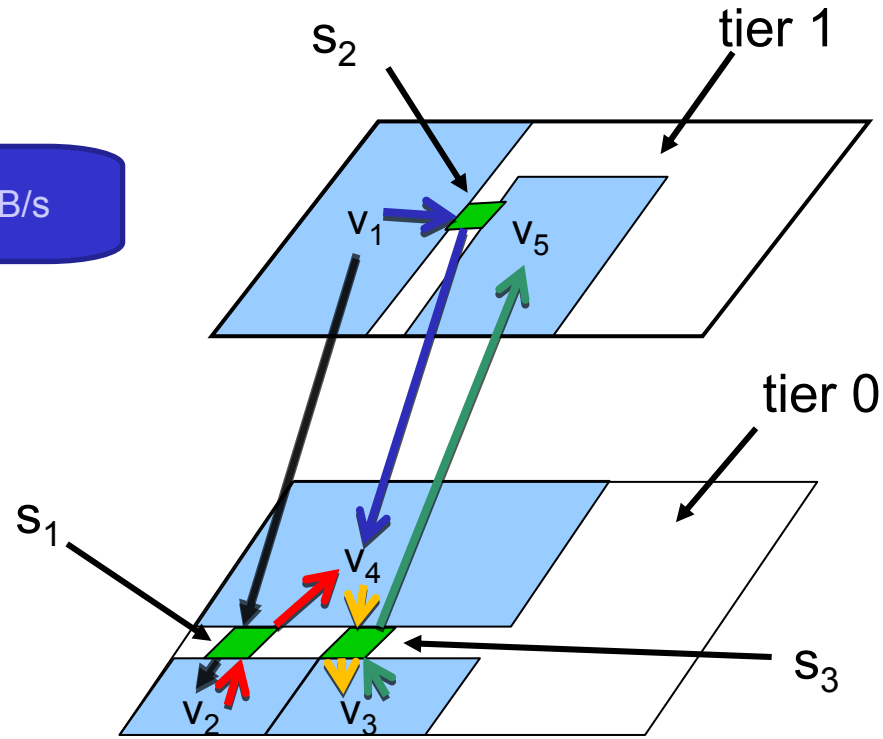


[Al-Sarawi, 1998]

A 3D Custom NoC Design Example



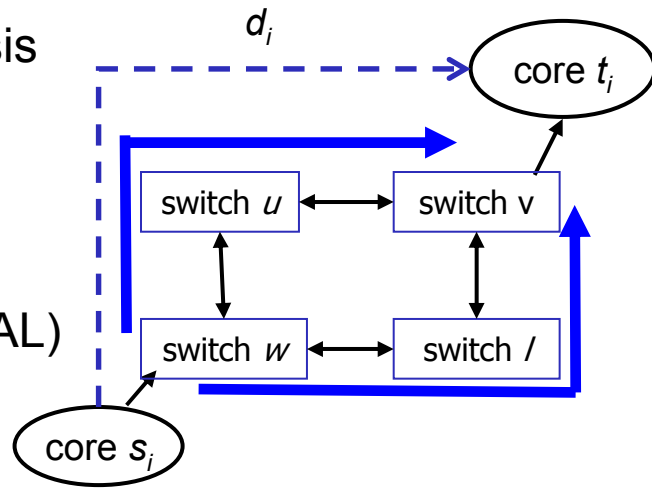
A given core graph



3D custom NoC architecture

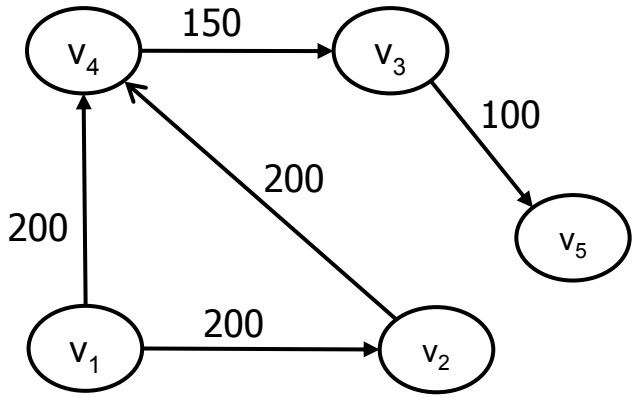
Our Solutions

- Drawbacks of prior work on custom NoC design
  - Fixed-order flow routing
    - performance depends on flow ordering
  - Hop-count as metric for NoC latency
    - more accurate models needed
  - No feedback on physical floorplan to NoC synthesis
- To overcome the routing order problem
  - Multi-commodity Flow (MCF) problem formulation
  - Solve MCF by Stochastic Simulated Allocation (SAL)
- Use accurate delay model for switches in NoCs
  - Queuing delay
  - Network contention
- Interleave NoC synthesis with floorplanning of cores/switches
  - Eliminate the dependence of NoC design on physical floorplan



Problem Formulation

- Input
  - Core graph (Communication Flow Graph)
  - Technology parameters
    - Clock frequency (in Hz)
    - Data link width (in bits)



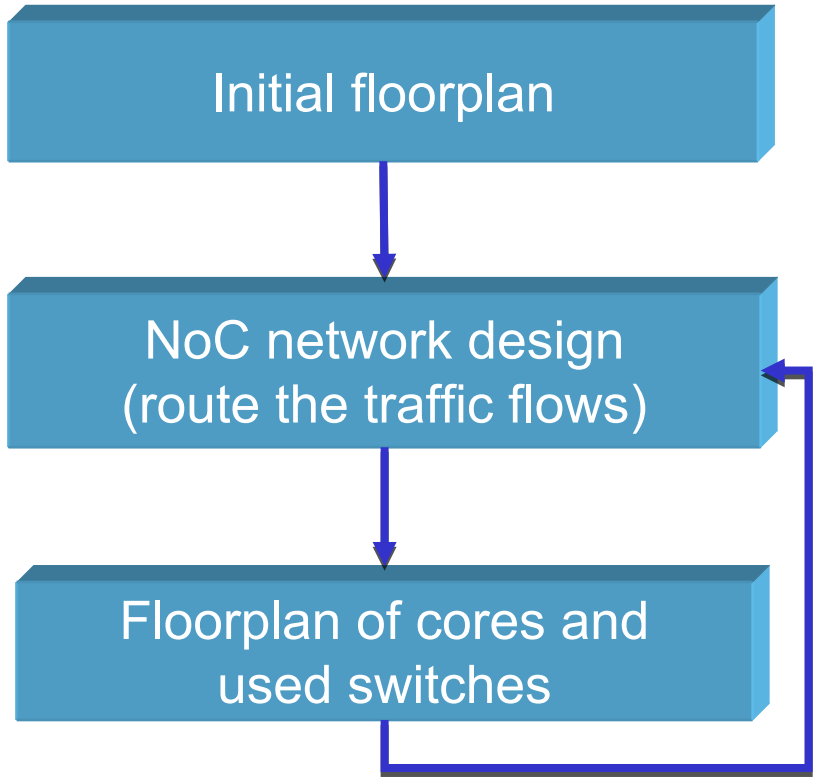
A given core graph

- Objectives
  - Floorplan and custom network topology design
    - Network topology, location of switches and # of ports of switches
  - Minimize network power (switch and link power), avg. packet latency , footprint size, TSV count and temperature

- Constraints
  - Bandwidth constraint on each physical link
  - Target frequency (link and switch delay must be within a value)

Algorithm Overview

- Initial floorplan of cores
  - Thermal-aware floorplanning
  - Optimize
    - Temperature
    - Sum of weighted link bandwidth
- NoC network design
  - Route each traffic flow in the core graph
  - Optimize
    - Network power
    - Network latency
    - TSV count
- Floorplan of cores + used switches
  - Find accurate physical position
  - Optimize link power and delay



Overall Flow

NoC Network Design (Routing)

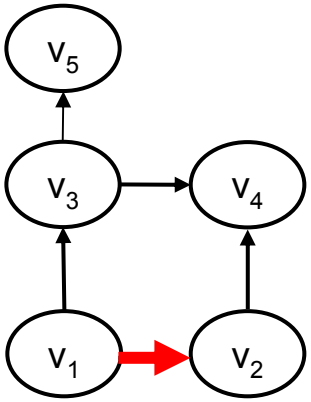
- Perform routing on routing graph
  - A routing graph represents all possible solutions
  - The collection of routing paths determines a network topology

- In routing graph, use a set of nodes to represent all usable switches in each tier
  - Proportional to the number of cores in a tier
  - Nodes from adjacent tiers form a complete graph

- Apply SAL algorithm to route all the traffic flows, which yields the NoC topology and the routes for the traffic flows in the routing graph

- Apply the turn-prohibition algorithm [Starobinski *et al*., TON, 2003] to guarantee deadlock-free routing

An Example

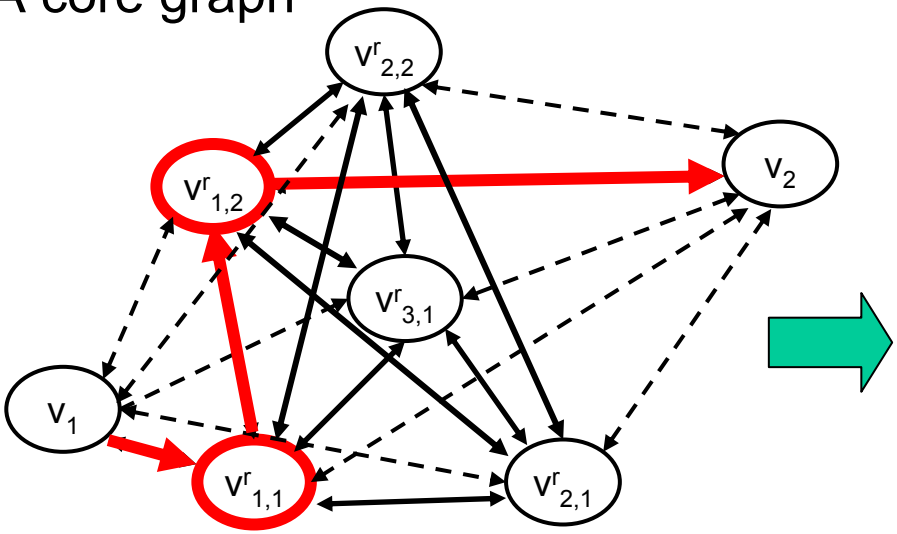


A core graph

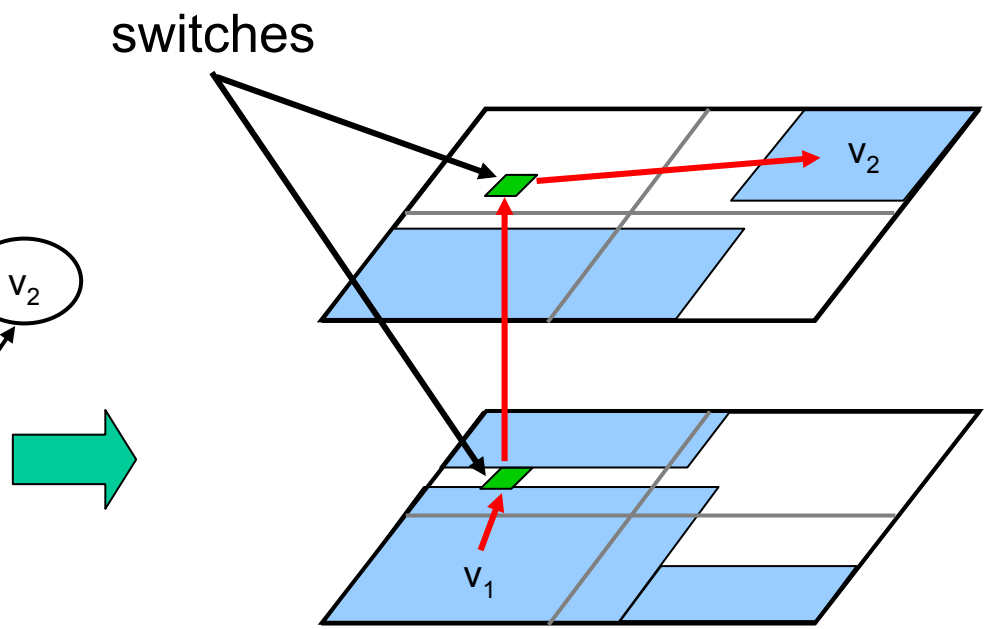
Five cores and two tiers

Red line represents a routing path

$V^r_{i,j}$: i-th switch node in tier j



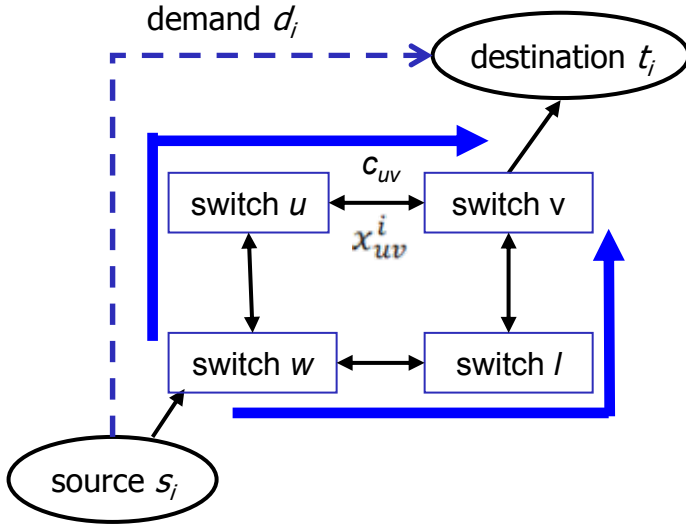
Routing graph



Floorplan

The Multi-commodity Flow (MCF) Problem

- Assume
  - K commodities (s_i, t_i, d_i) in the core graph,
  - c_{uv} : capacity on edge (u,v) in the routing graph
  - x_{uv}^i : flow of commodity i on edge (u,v)



- MCF problem is to find the optimal assignment of flow which satisfies the constraints:

Capacity constraints

$$\sum_{i=1}^{K} x_{uv}^i \leq c_{uv}$$

Supply/demand constraints

Flow conservation

$$\sum_{v} x_{uv}^i - \sum_{v} x_{vu}^i = \begin{cases} d_i & \text{if } u = s_i \\ -d_i & \text{if } u = t_i \\ 0 & \text{otherwise} \end{cases}$$

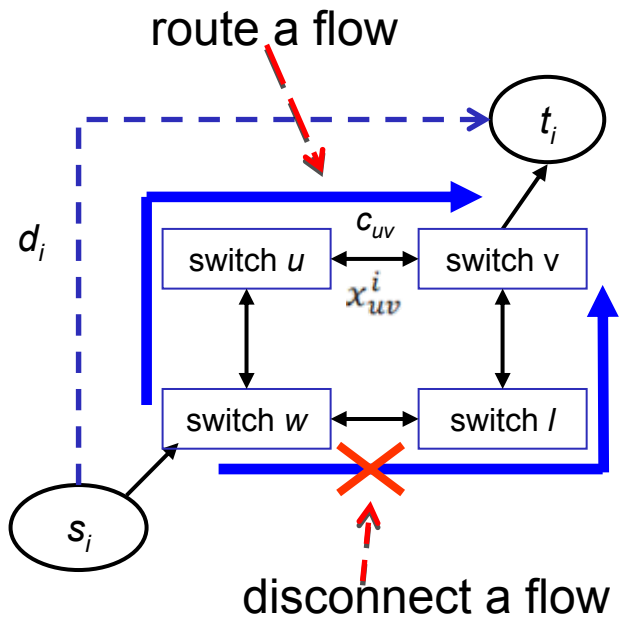
Simulated Allocation (SAL)

- A stochastic approach to solve the MCF problem
  - Simpler, but often faster and more efficient than simulated annealing and evolutionary algorithms
- SAL can deal with objective functions that are nonlinear or unavailable in closed form
- Use probability function $q(\gamma)$

$$\begin{cases} q(0) = 1 \\ q(H) = 0 \\ 1/2 < q(\gamma) \le 1, \ 0 < \gamma < H \end{cases}$$

to determine whether to route a traffic flow or to disconnect a flow

  - H : the total amount of traffic flow in the core graph



- Single-path routing for each traffic flow

Simulated Allocation (contd.)

Algorithm:

1: $n = 0$; $counter = 0$; $F^{best} = +\infty$; $x = 0$;
2 : repeat
3: if $random(0,1) < q(|x|)$ then
4. allocate(x);
5. else
6. disconnect(x);
7. end if
8. If $|x| = H$ then
9. $n = n + 1$; $counter = counter + 1$;
10. if $F(x) < F^{best}$ then
11. $F^{best} = F(x)$; $x^{best} = x$; $counter = 0$;
12. end if
13. end if
14. until $n = N$ or $counter = M$

x: allocation state, can be non-zero

route one randomly chosen non-allocated flow

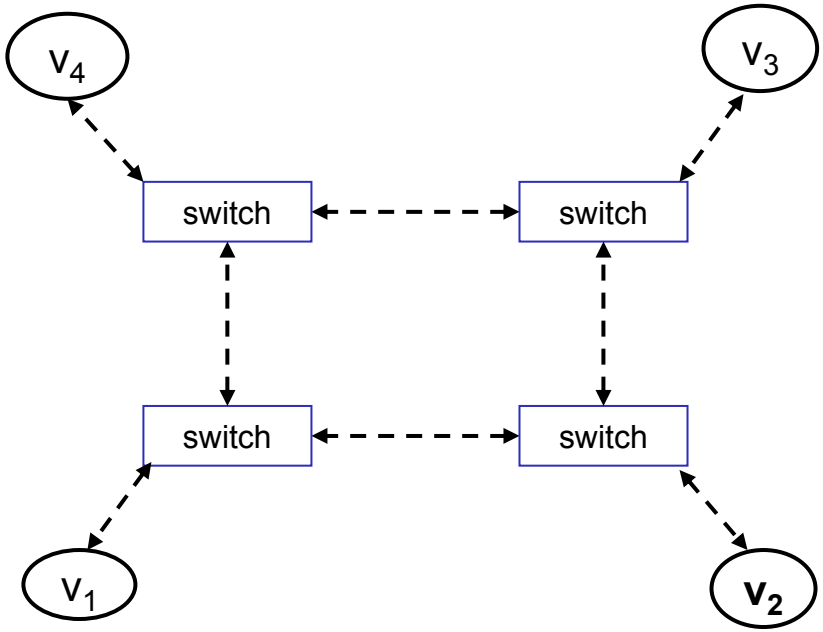
disconnect one randomly chosen allocated flow

all the flows are routed (full allocation state reached)

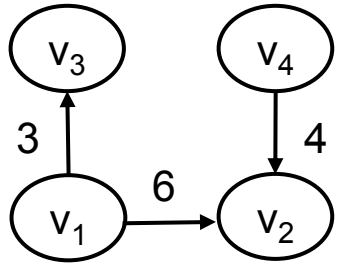
record new better solution

N, M: user-specified limits

An Example



Given routing graph
(incomplete)

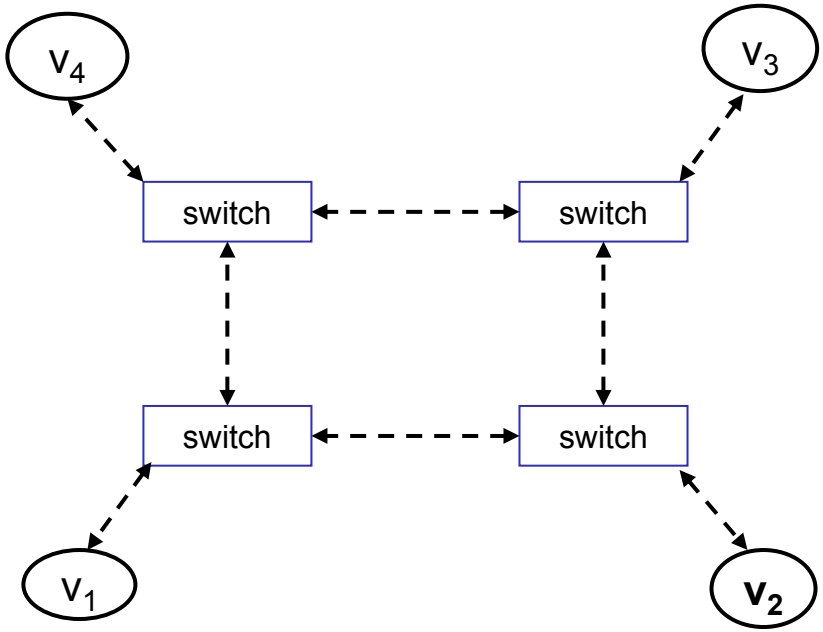


Given core graph (*H*=3)

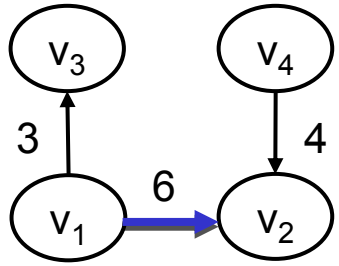
Assume identical link capacity: 10

An Example

random(0,1)<*q*(0)=1, route randomly chosen flow $v_1 \rightarrow v_2$

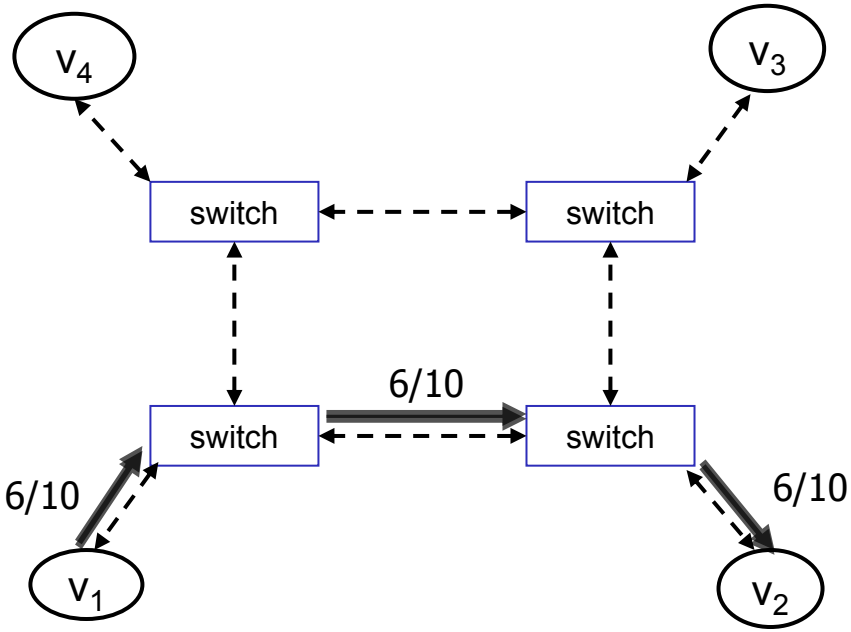


Given routing graph
(incomplete)

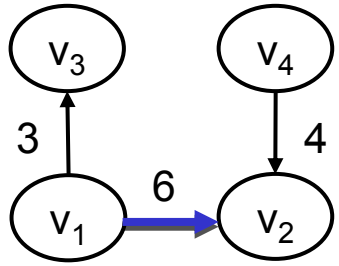


Given core graph (*H*=3)

An Example



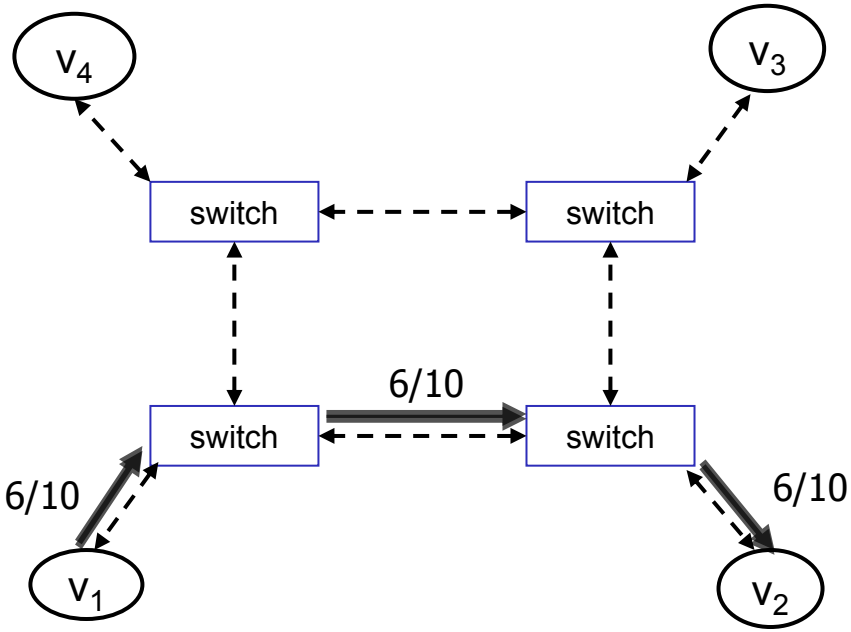
Given routing graph
(incomplete)



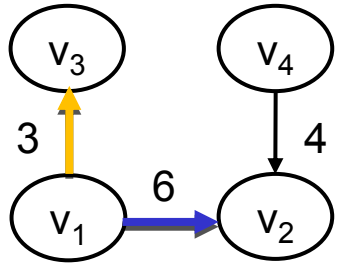
Given core graph (*H*=3)

An Example

Assume *random*(0,1)<*q*(1), route randomly chosen flow $v_1 \rightarrow v_3$

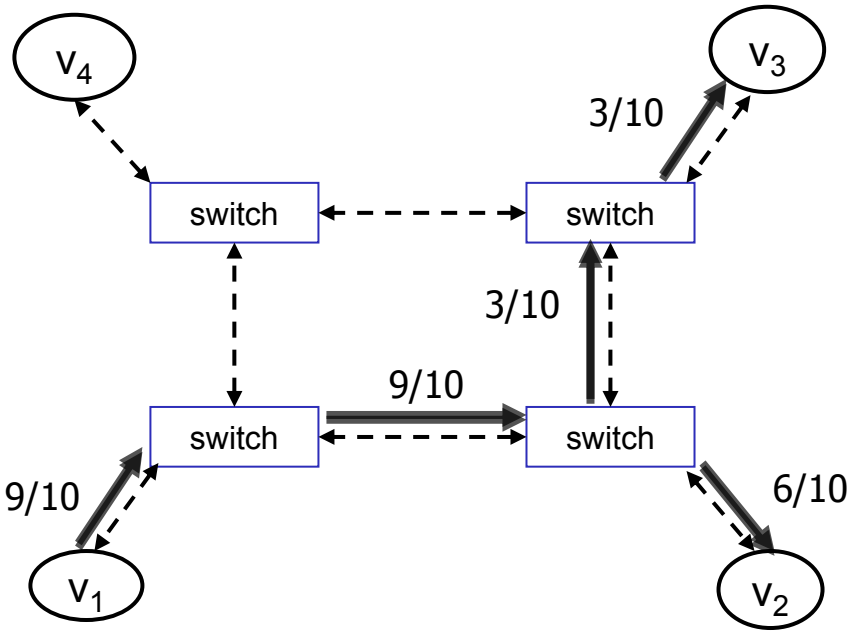


Given routing graph
(incomplete)

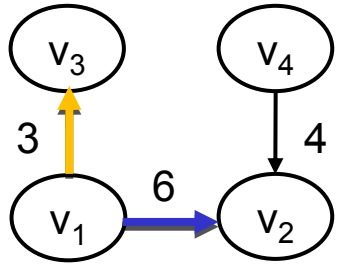


Given core graph (*H*=3)

An Example



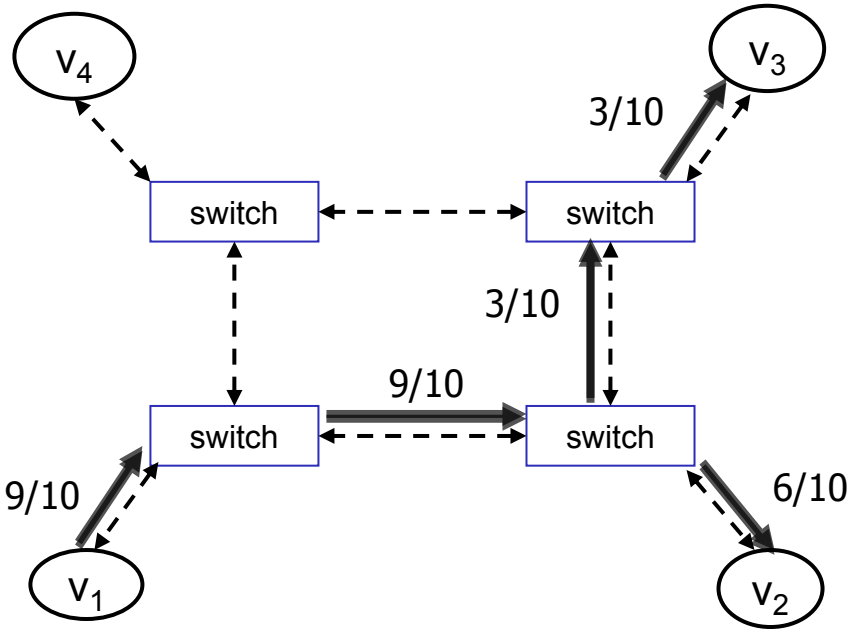
Given routing graph
(incomplete)



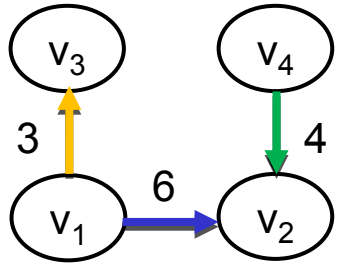
Given core graph (*H*=3)

An Example

Assume *random*(0,1)<*q*(2), route flow $v_4 \rightarrow v_2$

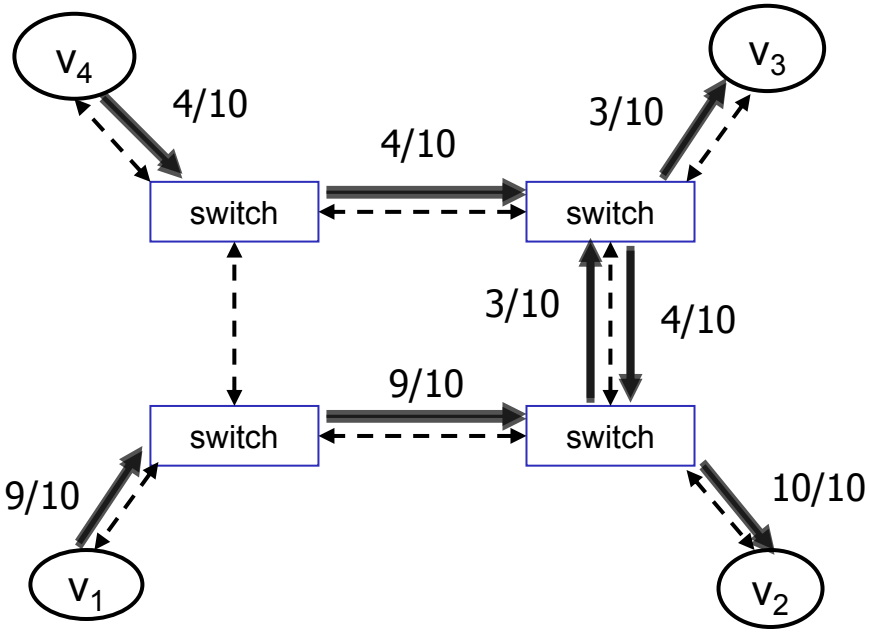


Given routing graph
(incomplete)

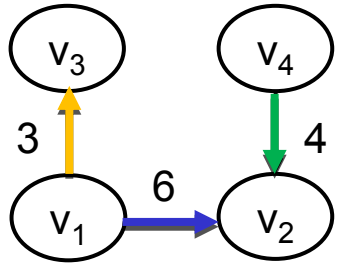


Given core graph (*H*=3)

An Example



Given routing graph
(incomplete)

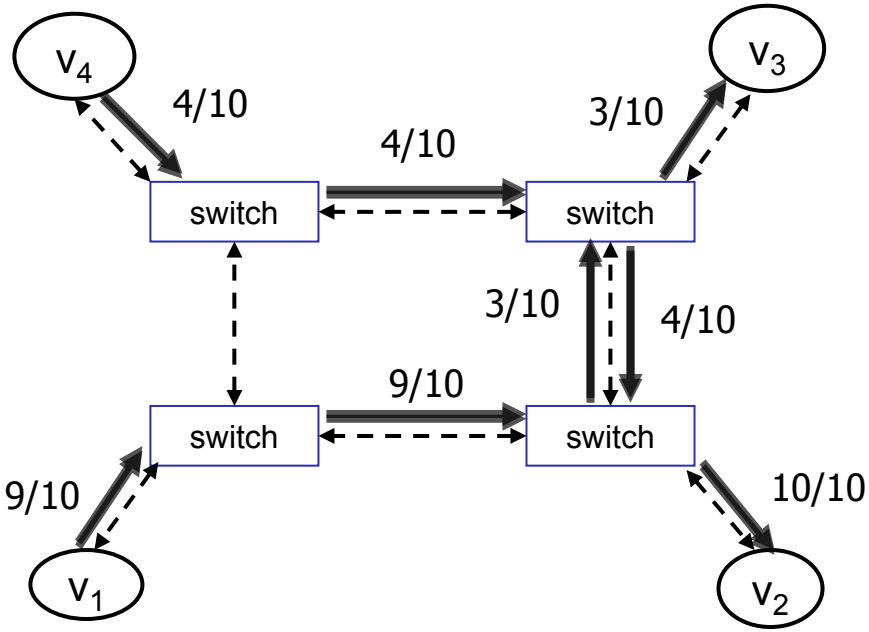


Given core graph (*H*=3)

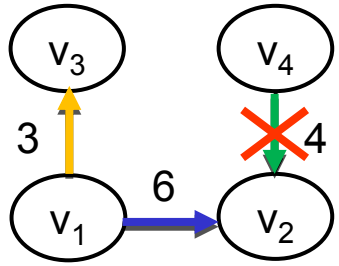
Reached one full allocation state !

An Example

random(0,1)>*q*(3)=0, disconnect randomly chosen flow $v_4 \rightarrow v_2$

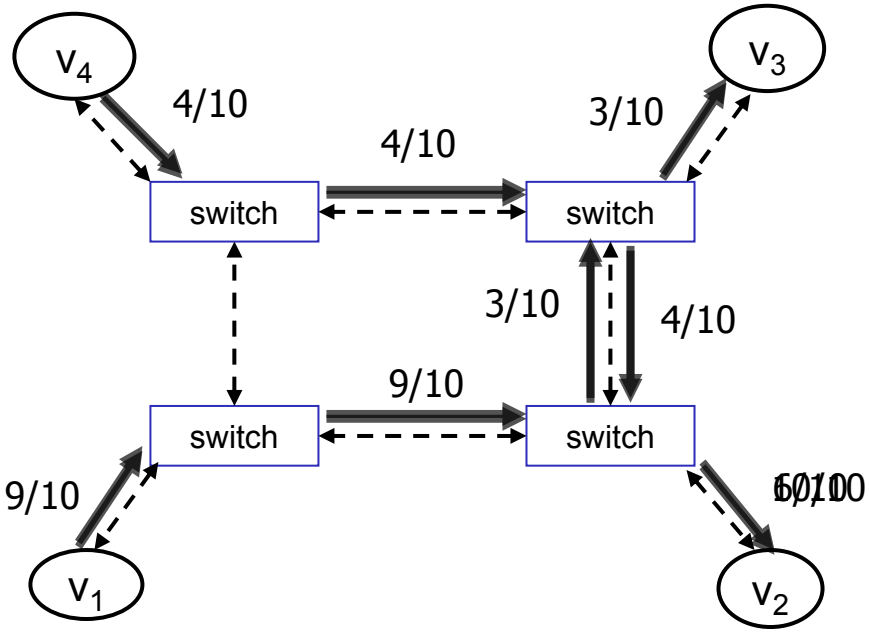


Given routing graph
(incomplete)

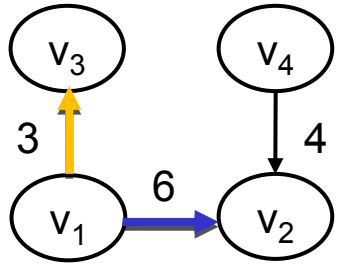


Given core graph (*H*=3)

An Example



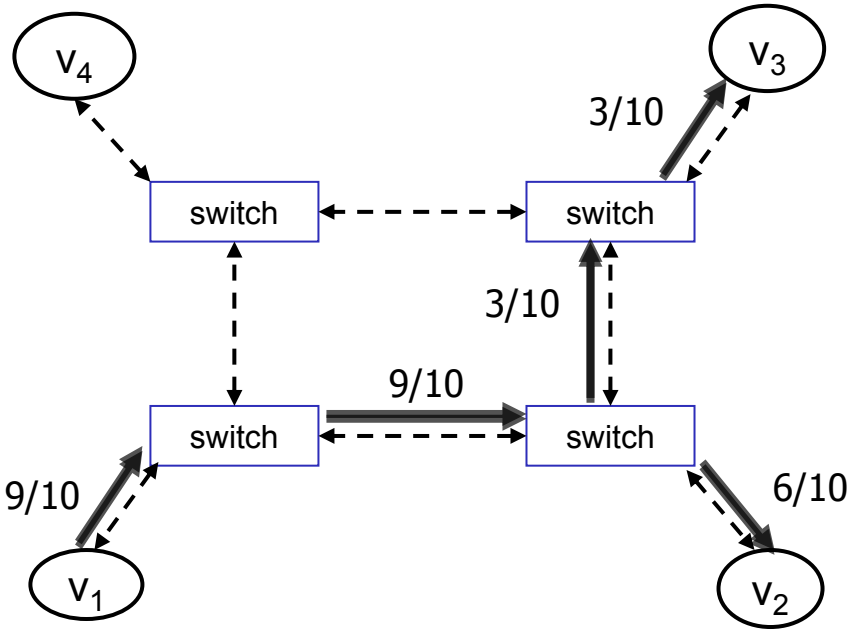
Given routing graph
(incomplete)



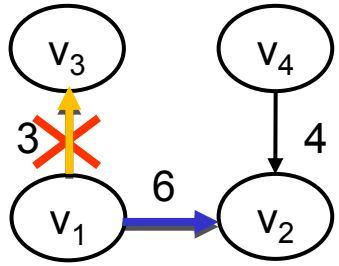
Given core graph (*H*=3)

An Example

Assume *random*(0,1)>*q*(2), disconnect randomly chosen flow $v_1 \rightarrow v_3$

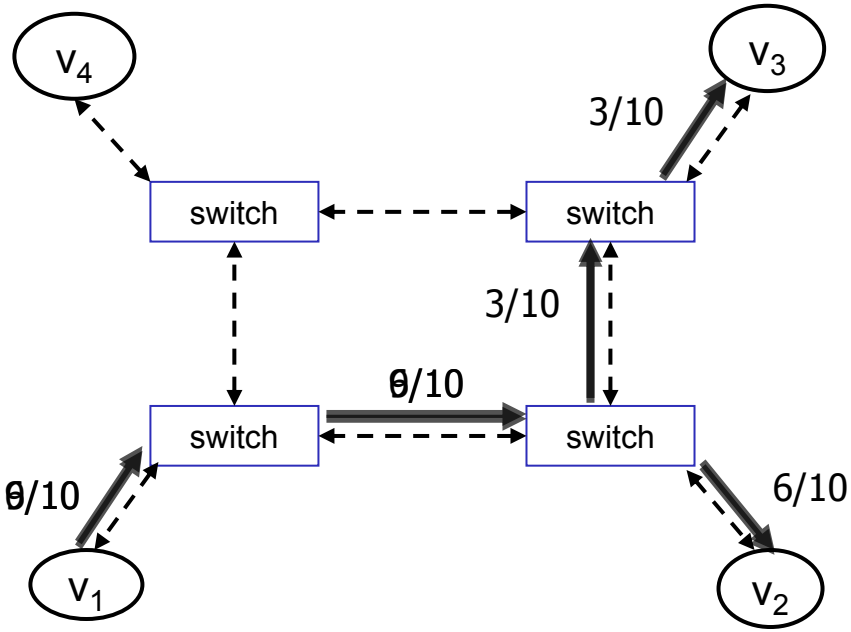


Given routing graph
(incomplete)

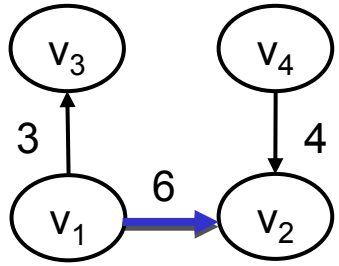


Given core graph (*H*=3)

An Example



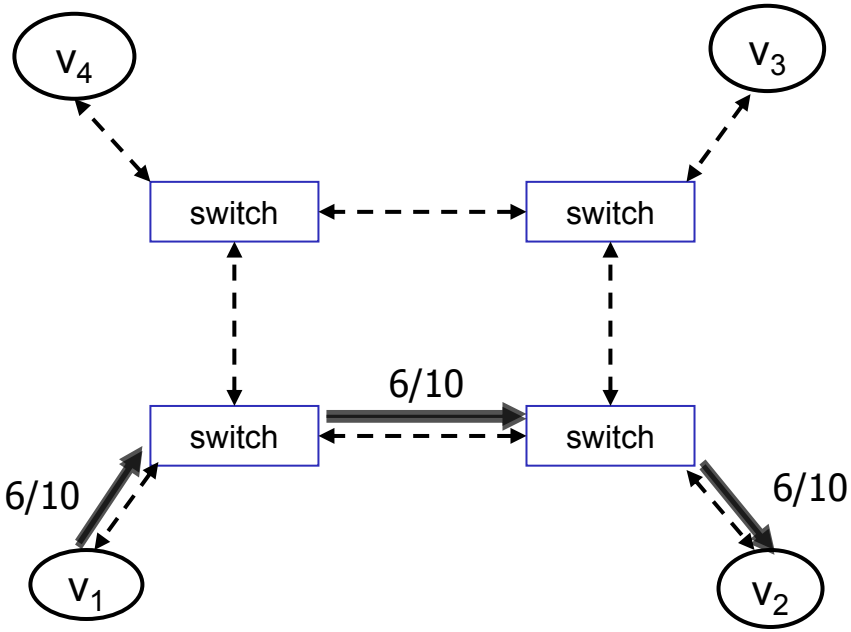
Given routing graph
(incomplete)



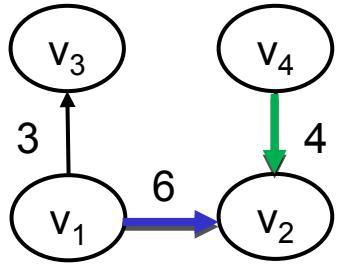
Given core graph (*H*=3)

An Example

Assume *random*(0,1)<*q*(1), route randomly chosen flow $v_4 \rightarrow v_2$

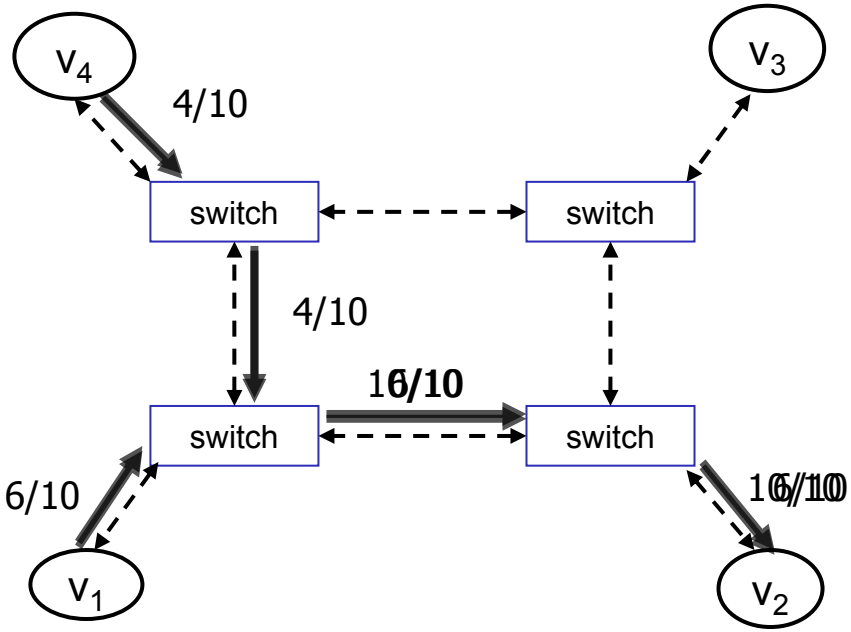


Given routing graph
(incomplete)

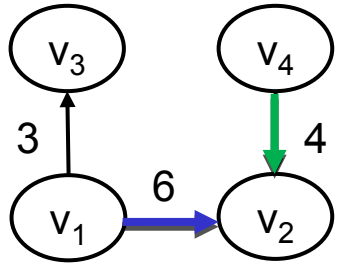


Given core graph (*H*=3)

An Example



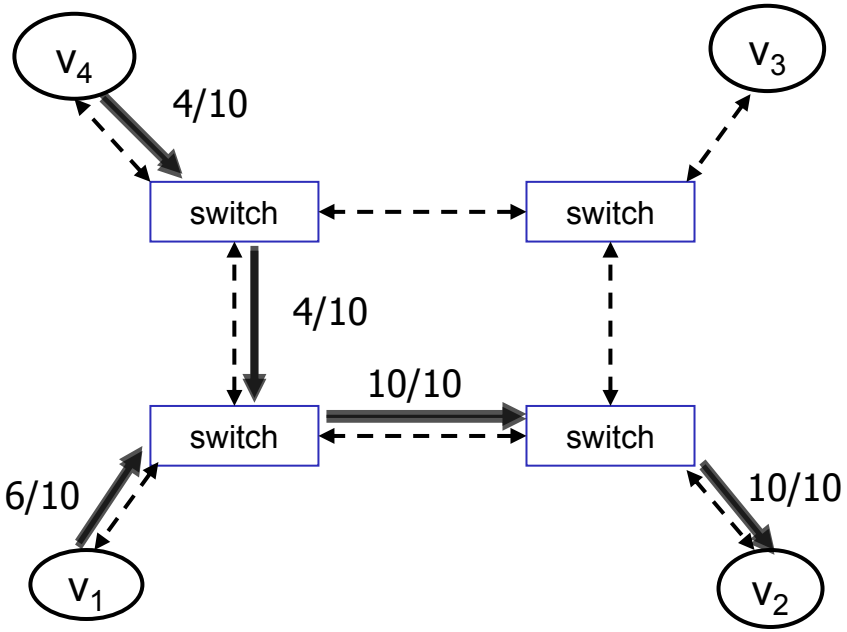
Given routing graph
(incomplete)



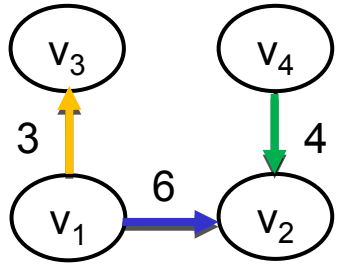
Given core graph (*H*=3)

An Example

Assume *random*(0,1)<*q*(2), route flow $v_1 \rightarrow v_3$

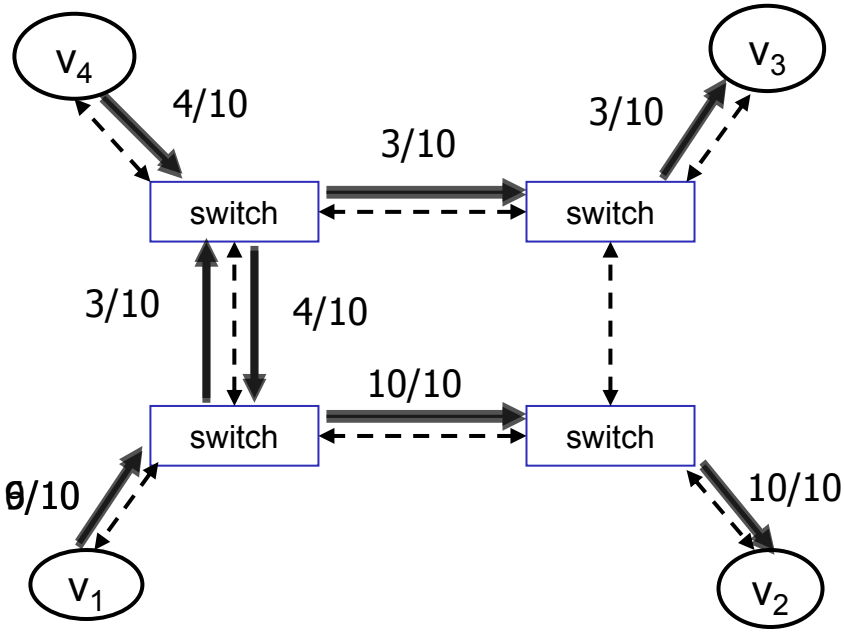


Given routing graph
(incomplete)

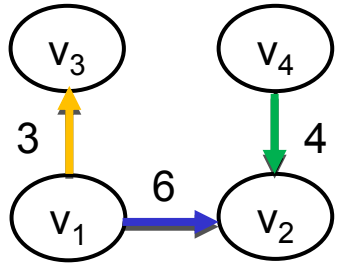


Given core graph (*H*=3)

An Example



Given routing graph
(incomplete)

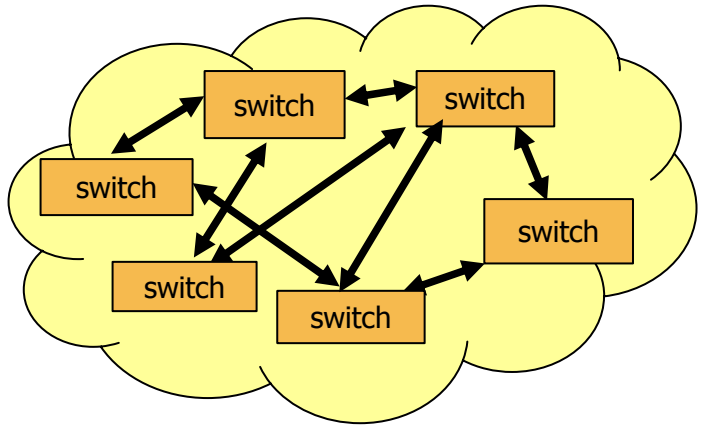


Given core graph (*H*=3)

Reached another full allocation state !

Models for NoC Components

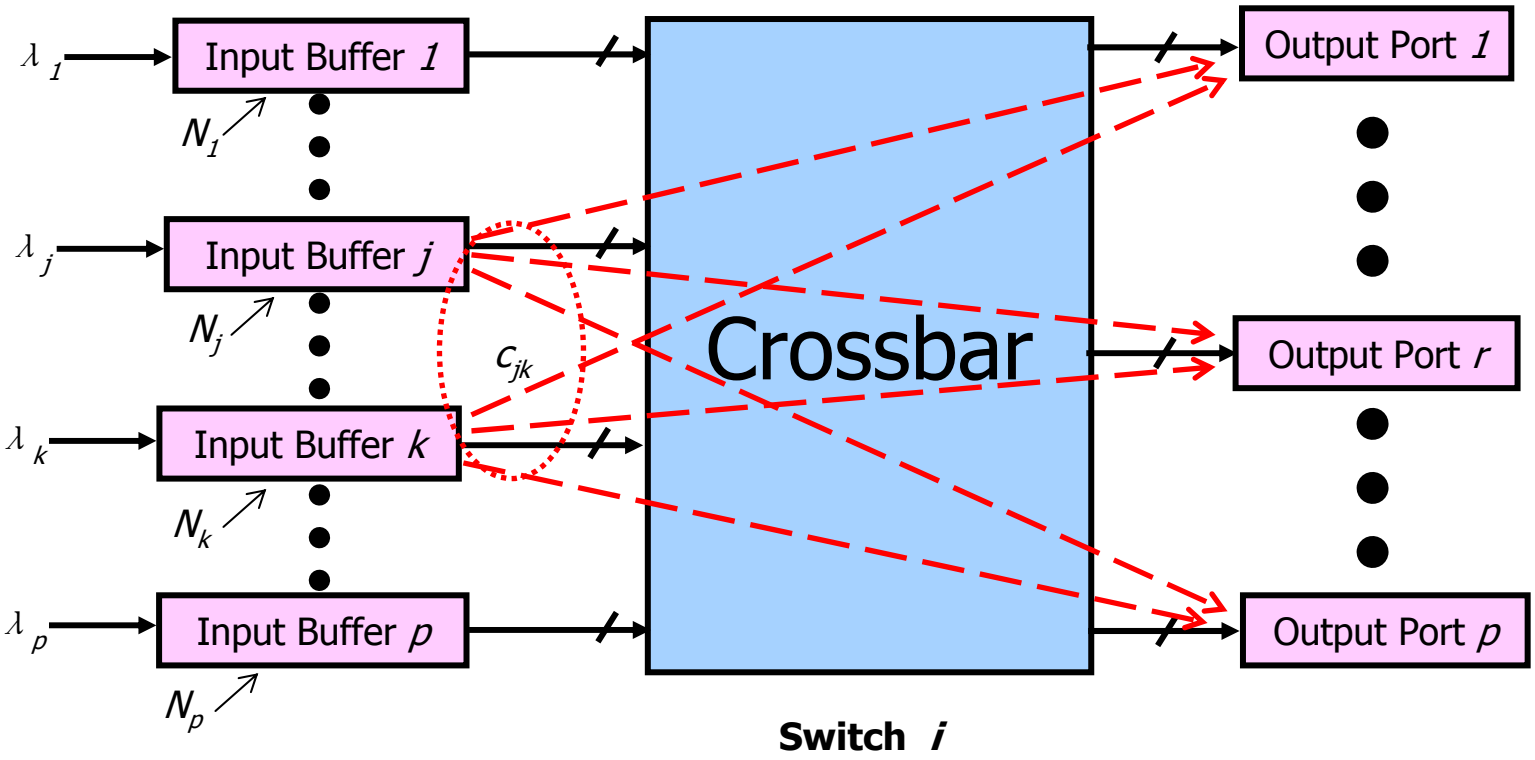
- NoC components: switch and link
- 45nm technology node
- Area and Power models
  - ORION tool (Wang *et al*., MICRO, 2002)



NoC

- Delay model for physical links
  - Use the equations from [Pavlidis *et al*., TVLSI, 2007]
- Accurate delay model for switches
  - Adapted from [Ogras *et al*., DATE, 2007]
  - Considers queuing delay and network contention

Accurate Switch Delay Model



- The equilibrium condition for the switch *i* can be written as:

$$(I - T \cdot \Lambda \cdot C) \cdot N = \Lambda \cdot \overline{R}$$

  - A closed form expression for N_j, the average number of packets at each input port
  - Can be used to compute the average waiting time in each input port by Little's theorem as $w_j = N_j / \lambda_j$

Experimental Setup

- Clock frequency: 900 MHz
- Packet size: 512 bits
- Flit width: 32 bits
- Buffer size: 4 flits
- Benchmarks:
  - 5 small published, 2 tiers
  - 5 large synthetic, 4 tiers

Published	# Cores	#Flows	Synthetic	# Cores	#Flows
PIP	8	8	B1	56	196
MWD	12	13	B2	80	96
VOPD	12	15	B3	69	136
MPEG4	12	26	B4	114	396
IMP	27	96	B5	124	266

Experimental Results

- Effect of each strategy on small published benchmarks

Ben	Baseline1		Baseline2		Baseline3		3D-SAL-FP	
	Power (mW)	Delay (ns)	Power (mW)	Delay (ns)	Power (mW)	Delay (ns)	Power (mW)	Delay (ns)
PIP	59	3.8	48	3.7	43	3.6	42	3.2
MWD	102	4.1	81	4.0	71	3.8	71	3.5
VOPD	110	7.3	92	7.2	83	6.9	81	5.1
MPEG4	180	10.3	123	10.1	100	9.0	103	6.3
IMP	702	9.4	512	8.0	422	7.8	425	6.4
	1	1	0.77	0.95	0.66	0.91	0.66	0.74

- Baseline1: fixed order routing + no floorplan feedback + simple delay model
- Baseline2: SAL traffic routing + no floorplan feedback + simple delay model
- Baseline3 : SAL traffic routing + floorplan feedback + simple delay model
- 3D-SAL-FP: SAL traffic routing + floorplan feedback + accurate switch delay model

Experimental Results (contd.)

- Effect of each strategy on large synthetic benchmarks

Ben	Baseline1		Baseline2		Baseline3		3D-SAL-FP	
	Power (mW)	Delay (ns)	Power (mW)	Delay (ns)	Power (mW)	Delay (ns)	Power (mW)	Delay (ns)
B1	1324	16.3	1258	16.0	1017	15.0	999	6.7
B2	911	7.9	689	7.9	589	7.6	590	4.6
B3	1076	13.1	737	12.0	674	11.5	645	9.4
B4	3955	15.9	3118	15.5	2469	13.9	2353	7.3
B5	2675	13.9	2203	11.8	1784	11.4	1806	9.1
	1	1	0.79	0.94	0.66	0.89	0.65	0.56

- Baseline1: fixed order routing + no floorplan feedback + simple delay model
- Baseline2: SAL traffic routing + no floorplan feedback + simple delay model
- Baseline3 : SAL traffic routing + floorplan feedback + simple delay model
- 3D-SAL-FP: SAL traffic routing + floorplan feedback + accurate switch delay model

Experimental Results (contd.)

- Delay and power reduction potential in 3D NoCs (B3)

Layers	Footprint (mm^2)	Link Power (mW)	Max path length (mm)	Max link delay (ns)	Max network latency (ns)	Avg network latency (ns)	# of TSVs	T_{max} (℃)
1	**216.8**	**288.4**	**22.1**	**6.45**	**14.40**	**12.42**	**0**	**43.8**
2	**110.3**	**189.2**	**17.0**	**4.95**	**12.28**	**9.56**	**86**	**63.7**
3	**72.0**	**164.8**	**11.9**	**3.50**	**11.51**	**9.49**	**94**	**96.2**
4	**56.1**	**141.0**	**9.2**	**2.68**	**11.32**	**9.44**	**116**	**118.0**

1. Increasing the number of 3D tiers results in smaller footprint size at the cost of more TSVs and higher chip temperature
2. Increasing the number of 3D tiers brings down max path length
   $\Rightarrow$ reduced link latency and power

Summary

- Propose an efficient application-specific 3D NoC design approach
- Use Multi-commodity Flow (MCF) problem formulation to overcome the routing order problem
- Solve MCF by Stochastic Simulated Allocation (SAL) approach
- Use accurate delay model for switches in NoCs which considers queuing delay and network Contention
- Interleave NoC synthesis with floorplanning of cores/switches to eliminate the dependence of NoC design on physical floorplan

Thank You!