EE 579: Wireless and Mobile Networks Design & Laboratory

Lecture 10

Amitabha Ghosh
Department of Electrical Engineering
USC, Spring 2014

Lecture notes and course design based upon prior semesters taught by Bhaskar Krishnamachari and Murali Annavaram.

Outline

- Administrative Stuff
- Traffic Management in Data Centers using Software Defined Networks (SDN)
- Puzzles

Scalable Multi-Class Traffic Management in Data Center Backbone Networks

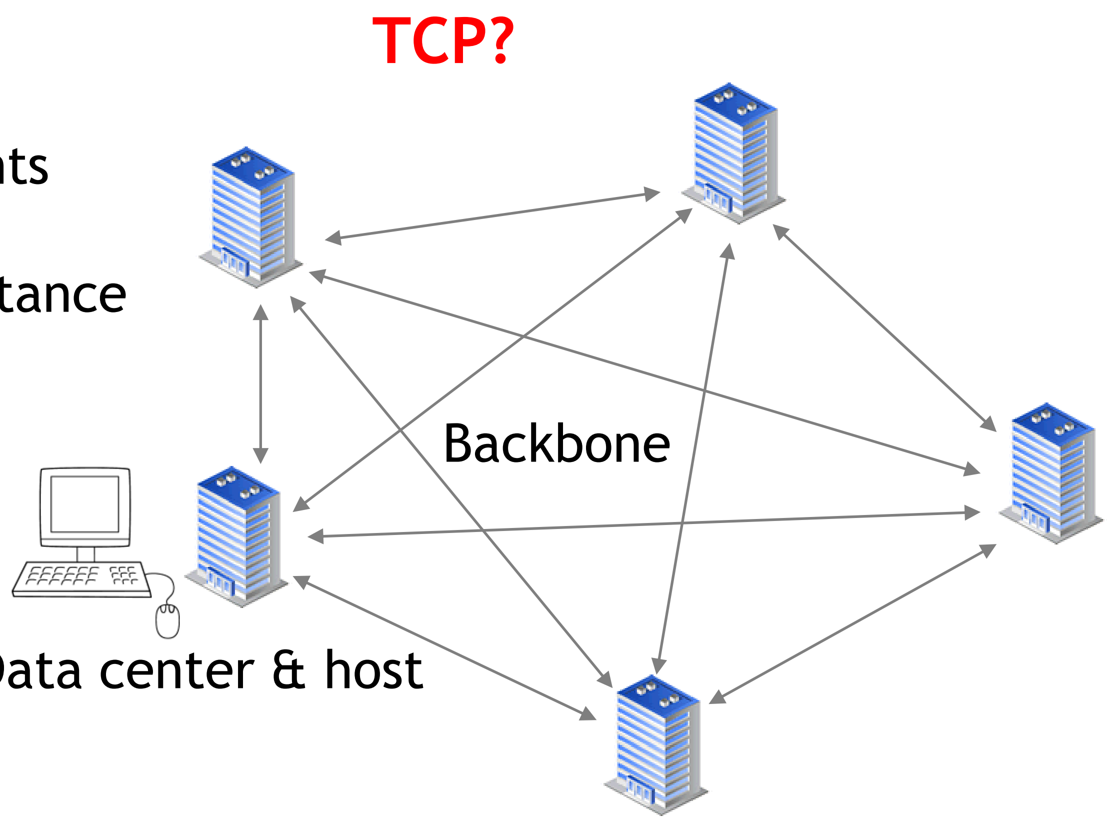
(Collaborators: Google, Princeton)

Outline

- Motivation
- Contributions
- Model and Formulation
- Scalable Designs
- Performance Evaluation
- Conclusions

Motivations

- Multiple interconnected data centers (DCs) with multiple paths between them
- DCs, traffic sources, and backbone owned by the same OSP, e.g., Google, Yahoo, Microsoft
- Traffic with different performance requirements
- Different business importance

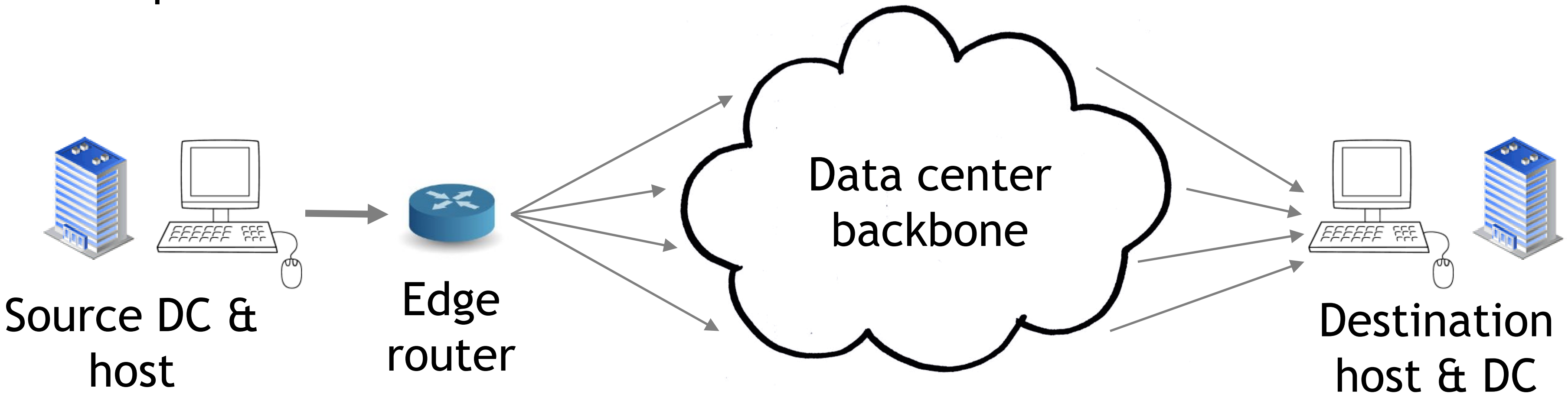


Contributions

Controlling the three “knobs”

- ❑ Sending rates of hosts
- ❑ Weights on link schedulers
- ❑ Splitting of traffic across paths

Joint optimization of rate control, routing, and link scheduling



Contributions

TRUMP (CoNEXT ‘07)
DaVinci (CoNEXT ‘08)

Fully-centralized
Not scalable

Semi-centralized: Our work
Modular, scalable with low message-passing and fast convergence

Fully-distributed
Scaling issues due to message passing, slow convergence

- Computation is distributed across multiple tiers using a few controllers
- Result is provably optimal using optimization decomposition
- Semi-centralized solutions viable and, in fact, preferred in practice, e.g., Google’s B4 globally-deployed software defined private WAN (SIGCOMM ‘13)

Model and Formulation

Traffic Model

- Performance requirements ⇨ Set of traffic classes $\mathcal{K} = \{k\}$
- Multiple flows per class
  - Flow: traffic between a source-destination pair $s \in \mathcal{F}^k$
- Business importance ⇨ flow weight w_s^k

Utility Function of a Class

- All flows in the same class have the same utility function $U^k(\cdot)$
- For simplicity, assume only throughput and delay sensitive traffic $f^k(\cdot)$ $g^k(\cdot)$

Model and Formulation

Network Model

- Set of unidirectional links $\mathcal{L} = \{l\}$
  - Capacity c_l
  - Propagation delay p_l
- Set of paths $\mathcal{P} = \{p\}$
- Routing matrix $\mathbf{A} = [A_{lp}]$ $\mathbf{R} = [R^k_{sp}]$

Topology matrix

smaller

Path routing matrix

larger

- One queue per class
- Multi-path routing
  - Path rate of flow s of class k over path p z^k_{sp}

Model and Formulation

Utility of Flow s of Class k

Coefficients to model different degrees of sensitivity to throughput and delay

$$U_s^k = w_s^k \left[a^k f^k(x_s^k) - b^k g^k(u_l^k) \right]$$

Weight of flow s of class k

Throughput sensitivity of class k, e.g., log(.)

Total sending rate of flow s of class k

Delay sensitivity of class k

Utilization of class k over link l

Sum of the products of path rates and average end-to-end delays on those paths

Model and Formulation

Objective Function

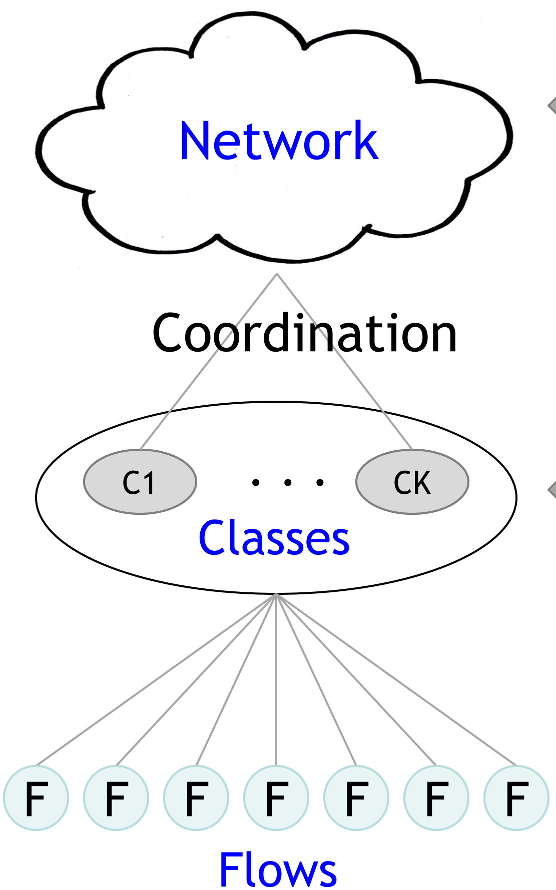
- Data centers, backbone and traffic sources under the same OSP ownership
- Maximize the sum of utilities of all flows across all traffic classes (global "social welfare")

Global Problem G:

$$
\begin{aligned}
\text{maximize} \quad & \mathcal{U} = \sum_k \sum_{s \in \mathcal{F}^k} U_s^k \\
\text{subject to} \quad & \mathbf{A}\mathbf{R}^k \mathbf{z}^k \preceq \mathbf{y}^k, \quad \forall k \\
& \sum_k y_l^k \leq c_l, \quad \forall l \\
\text{variables} \quad & \mathbf{z}^k \succeq 0, \quad \forall k
\end{aligned}
$$

Two-Tier Design

Each controller has a limited view about the network and inter-DC traffic



$\mathbf{y}^k$

Link Coordinator (LC)
Optimizes aggregate link bandwidth across classes

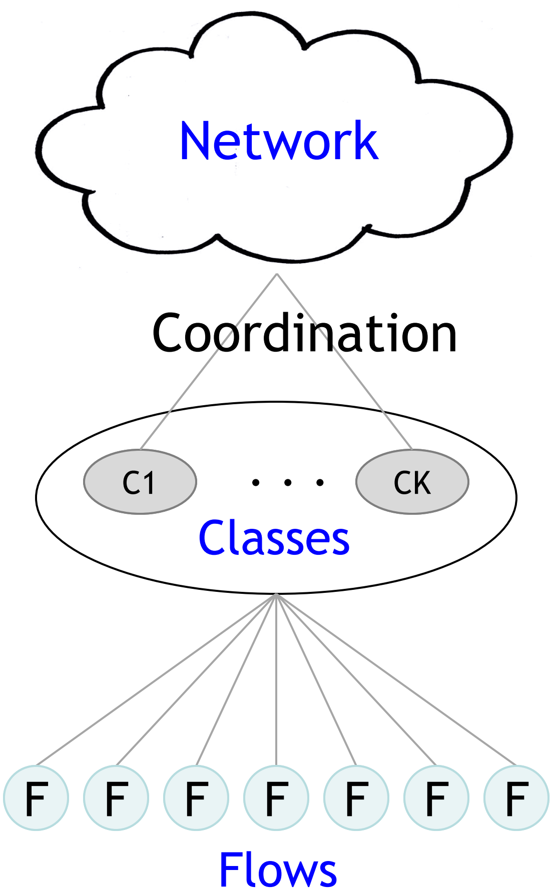
- A centralized entity
- Knows network topology

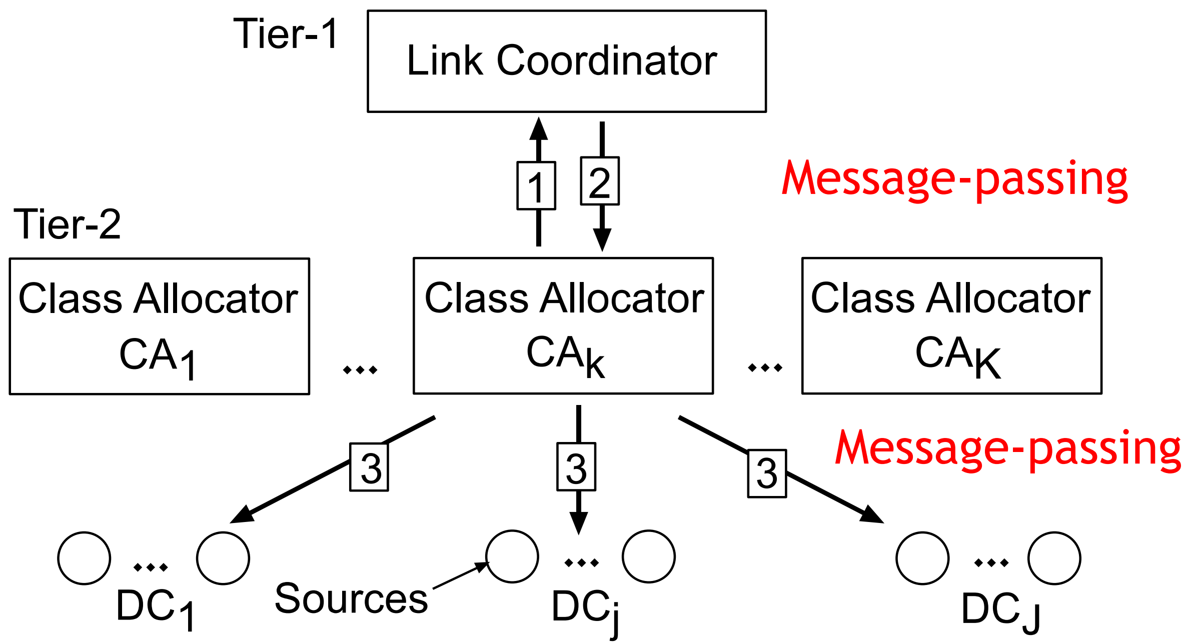
$\mathbf{z}^k$

Class Allocator (CA)
Optimizes sending rates across flows in its own class

- One for each class
- Knows the utility function, weights, and paths of individual flows in its own class

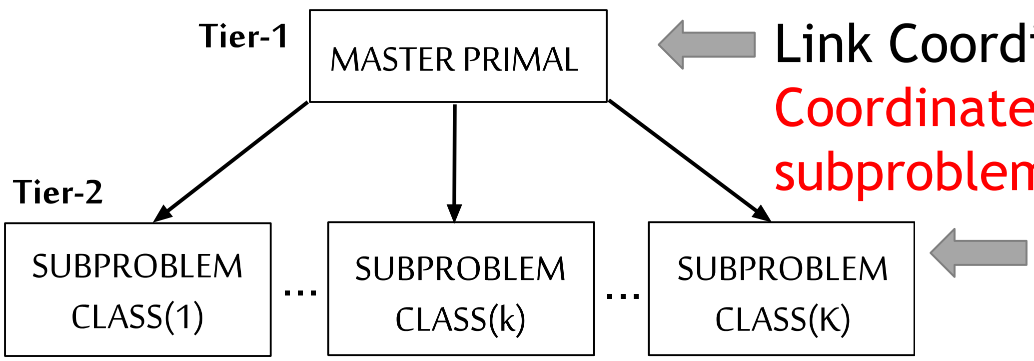
Two-Tier Design





Two-Tier Decomposition

Primal Decomposition

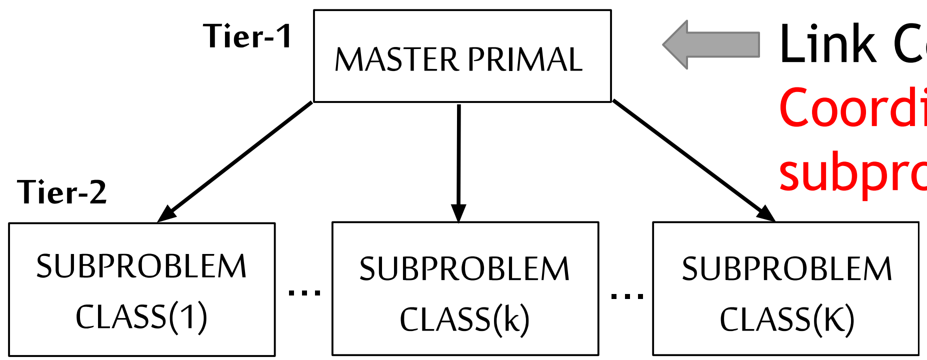


Link Coordinator
Coordinates all the subproblems

Class Allocator
Solves independently

Two-Tier Decomposition

Primal Decomposition



Link Coordinator
Coordinates all the subproblems

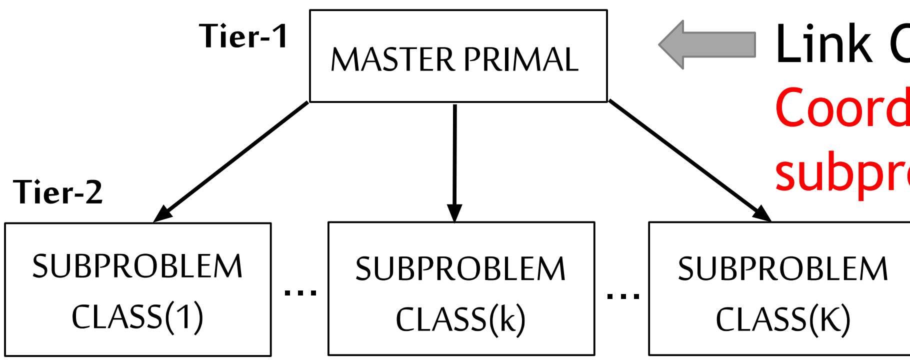
Class Allocator
Solves independently

Subproblem for Class k

$$\begin{aligned} \text{maximize} \quad & \mathcal{U}^k = \sum_{s \in \mathcal{F}^k} U_s^k \\ \text{subject to} \quad & \mathbf{A}\mathbf{R}^k \mathbf{z}^k \preceq \mathbf{y}^k \quad \forall k \\ \text{variables} \quad & \mathbf{z}^k \succeq 0 \end{aligned}$$

Two-Tier Decomposition

Primal Decomposition



Link Coordinator
Coordinates all the subproblems

Class Allocator
Solves independently

Subproblem for Class k

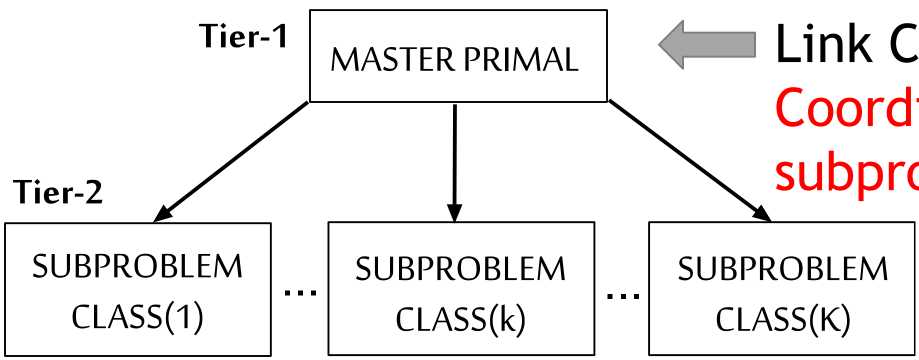
$$
\begin{array}{ll}
\text{maximize} & \mathcal{U}^k = \sum_{s \in \mathcal{F}^k} U_s^k \\
\text{subject to} & \mathbf{A}\mathbf{R}^k \mathbf{z}^k \preceq \mathbf{y}^k \quad \forall k \\
\text{variables} & \mathbf{z}^k \succeq 0
\end{array}
$$

Master Primal

$$
\begin{array}{ll}
\text{maximize} & \mathcal{U} = \sum_k \mathcal{U}^{k^*}(\mathbf{y}^k) \\
\text{subject to} & \sum_k y_l^k \le c_l \quad \forall l \\
\text{variables} & \mathbf{y}^k \succeq 0
\end{array}
$$

Two-Tier Decomposition

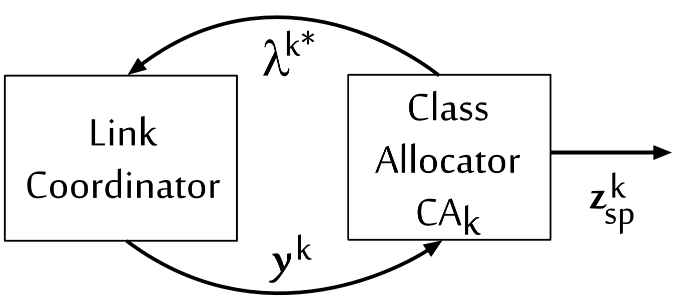
Primal Decomposition



Link Coordinator
Coordinates all the subproblems

Class Allocator
Solves independently

Message-Passing

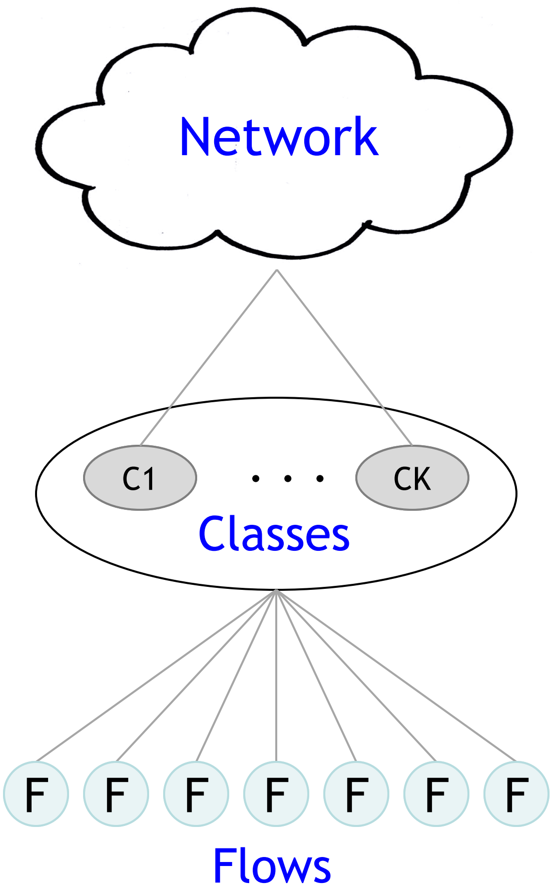


step size : β

$\mathbf{y}^k$: Aggregate bandwidth assigned to class k

$\boldsymbol{\lambda}^{k*}$: Optimal subgradient of CLASS(k)

Three-Tier Design



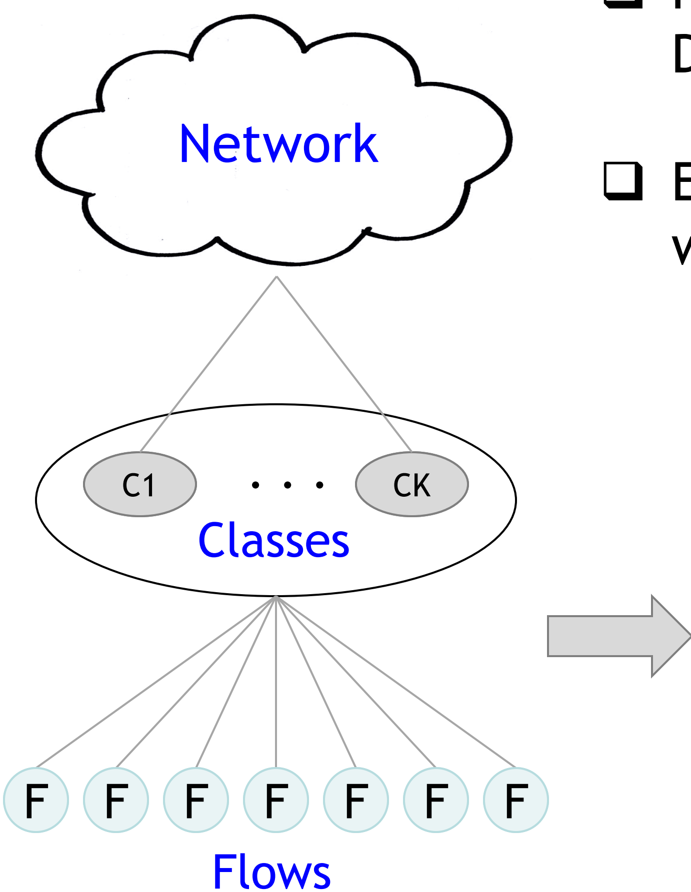
Why another tier? (High control overhead)

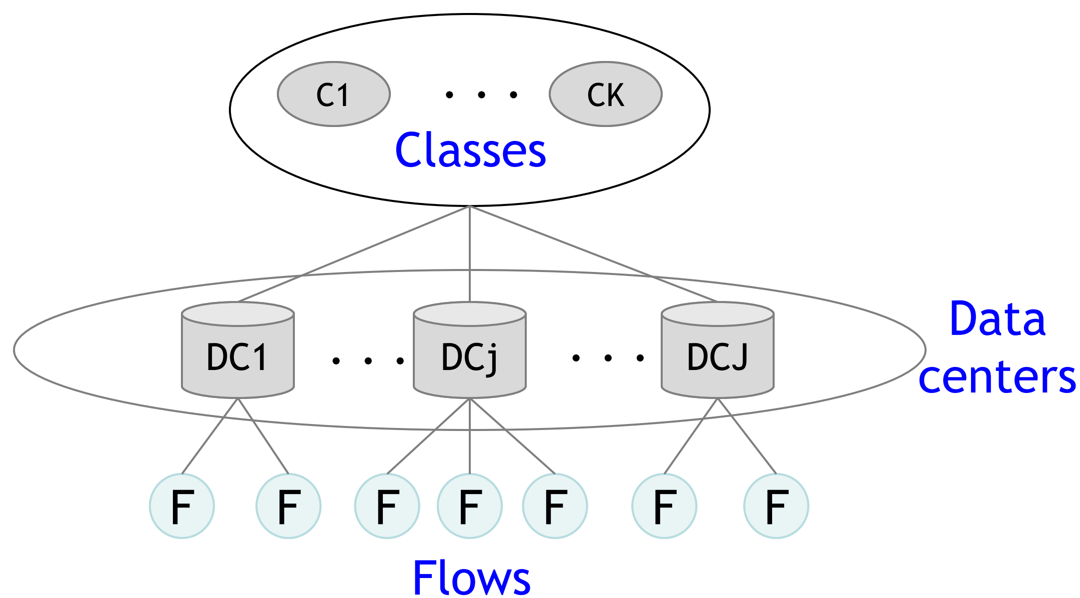
- Flow of a given class may originate from any DC
- Each class allocator potentially communicates with all DCs

Three-Tier Design

Why another tier? (High control overhead)

- Flow of a given class may originate from any DC
- Each class allocator potentially communicates with all DCs





Three-Tier Design

One centralized entity

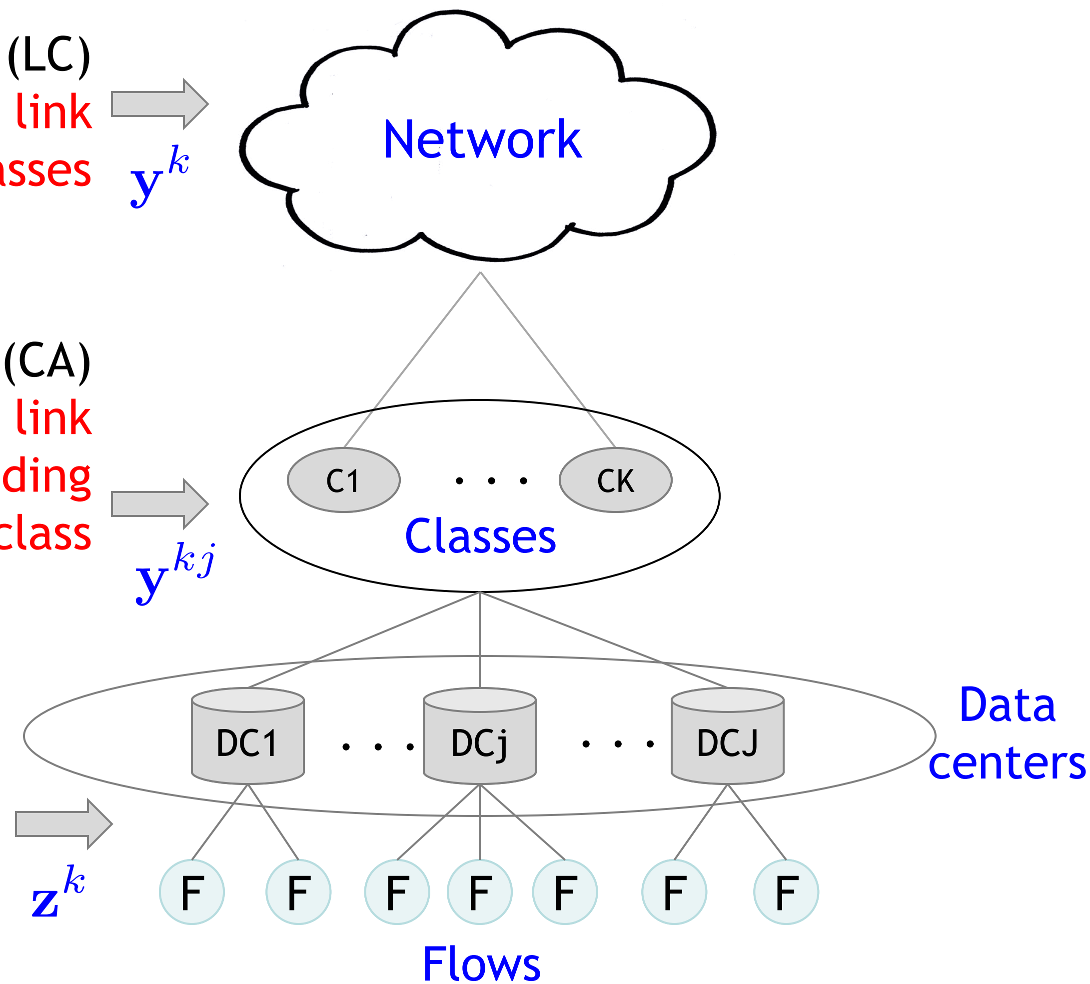
Link Coordinator (LC)
Optimizes aggregate link bandwidth across classes

One per class

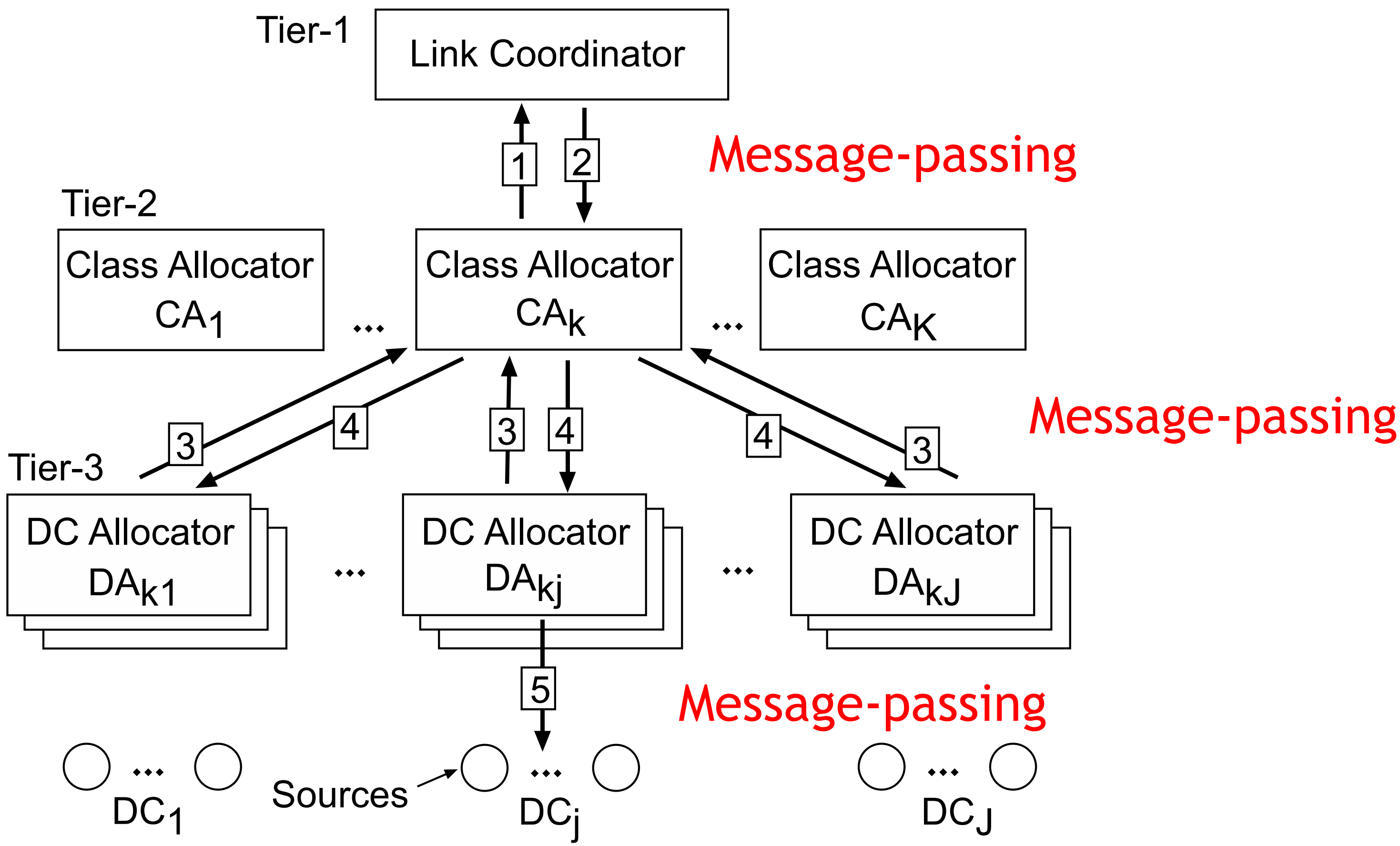
Class Allocator (CA)
Optimizes aggregate link bandwidth across DCs sending traffic in its own class

Data Center Allocator (DCA)
Optimizes sending rates across flows in its own class originating from its own DC

One per class, per DC

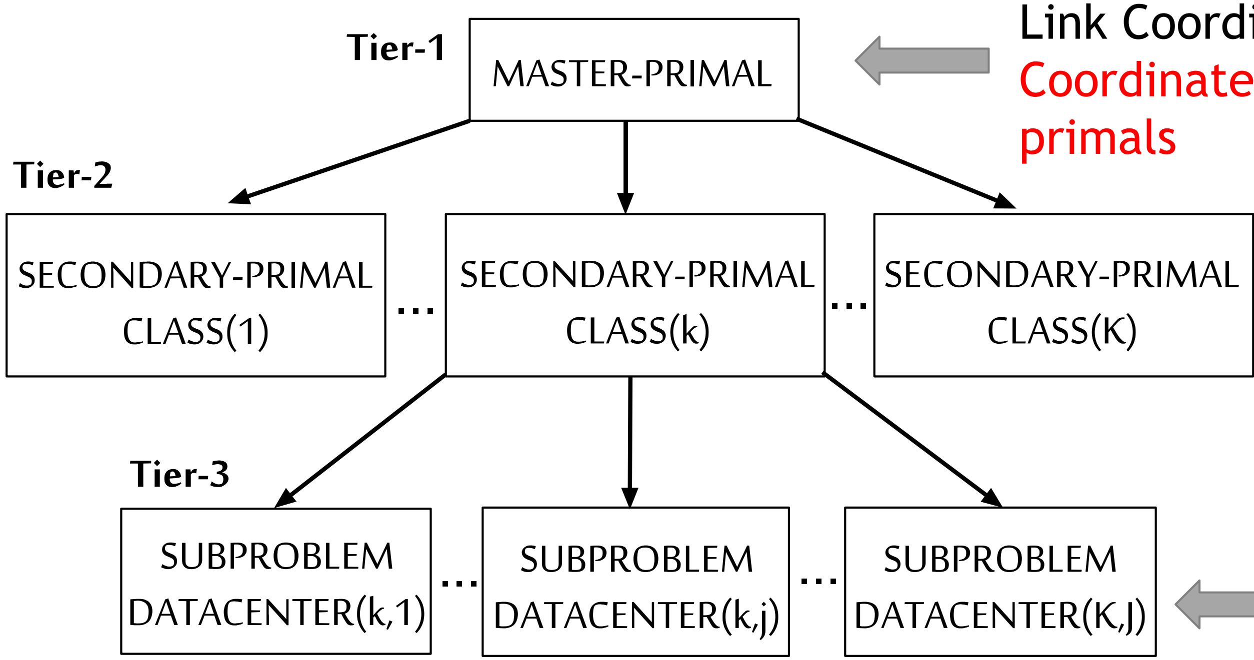


Three-Tier Design



Three-Tier Decomposition

2-Level Primal Decomposition



Link Coordinator
Coordinates the secondary primals

Class Allocator
Coordinates the subproblems of its own class

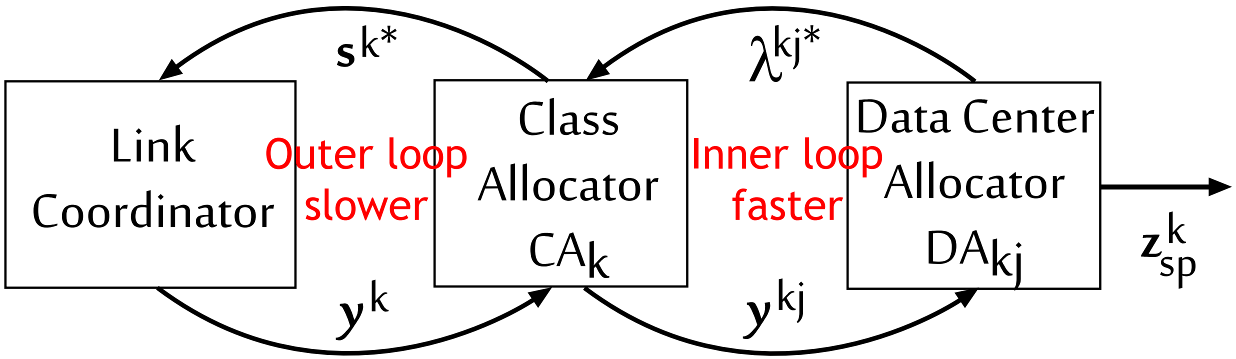
Data Center Allocator
Solves independently

Three-Tier Decomposition

Message-Passing

$\mathbf{s}^{k^*}$: Optimal subgradient of CLASS(k)

$\boldsymbol{\lambda}^{kj^*}$: Optimal subgradient of DATACENTER(k,j)



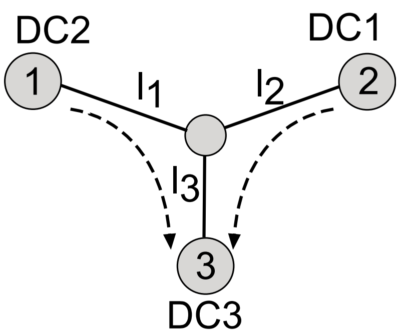
$\mathbf{y}^k$: Aggregate bandwidth assigned to class k

$\mathbf{y}^{kj}$: Aggregate bandwidth assigned to DC j sending traffic of class k

Performance Evaluation

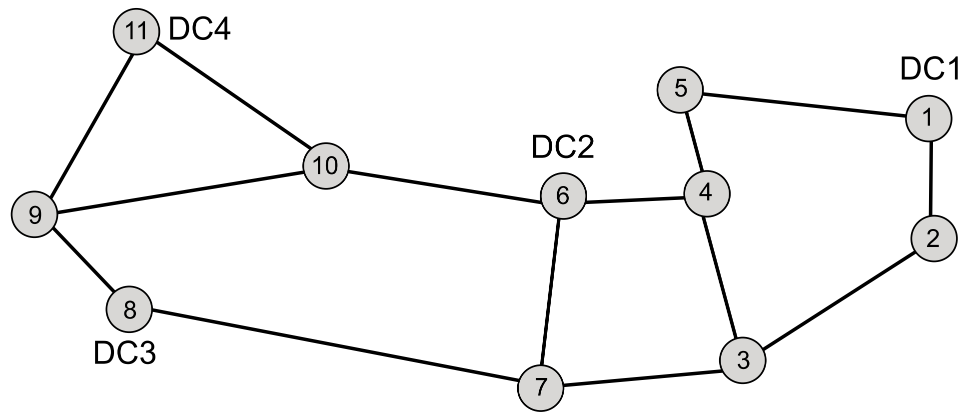
Performance Metrics

- ❑ Rate of convergence
- ❑ Message-passing overhead



Simple topology

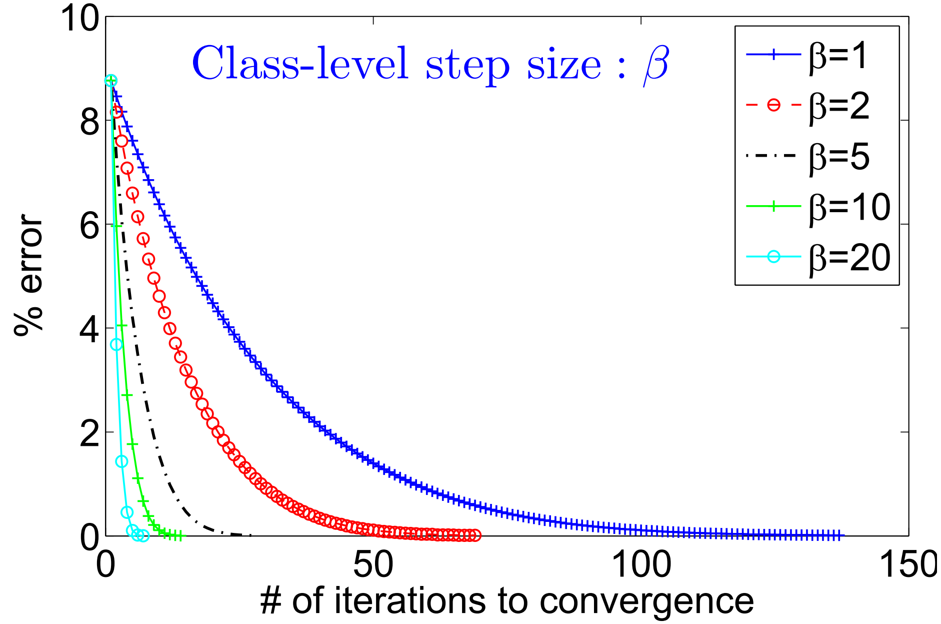
- DC1 & DC2 send traffic to DC3
- 100 Mbps link capacity
- Two classes with log utility



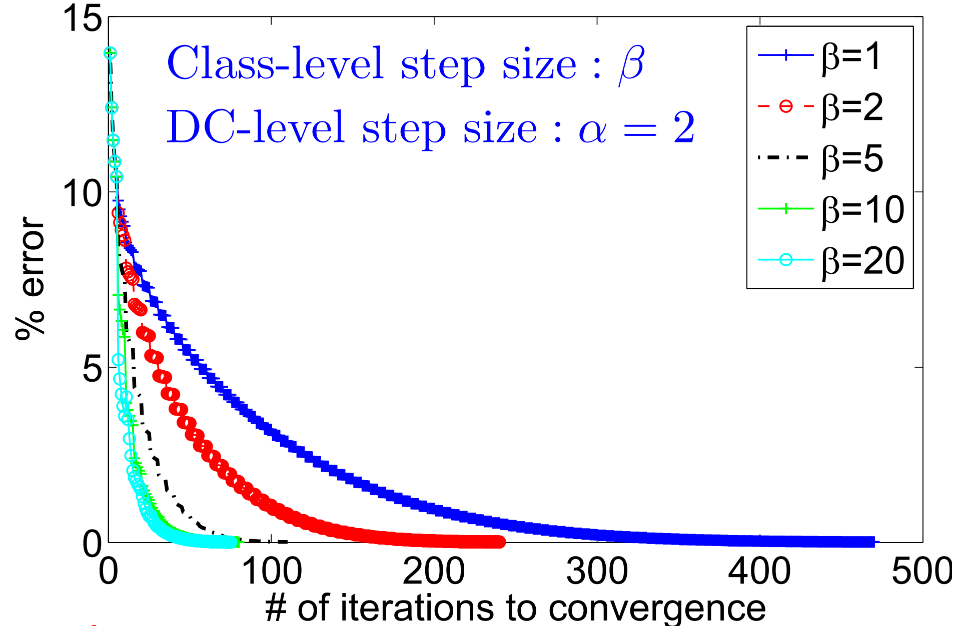
Abilene topology

- 4 DCs
- 1 Gbps link capacity in each direction
- First 3 shortest possible paths between every pair of DCs (36 total)
- Two classes with log utility functions

Rate of Convergence



Two-tier

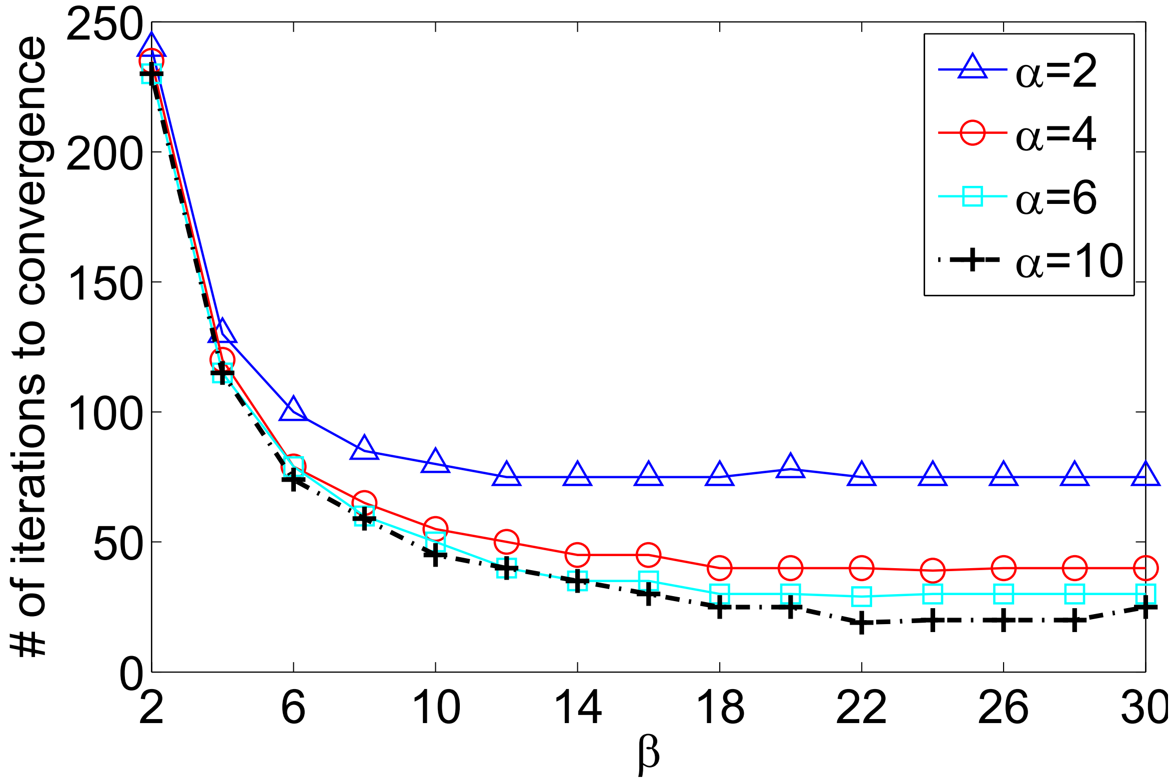


Three-tier

Each iteration is in the order of a few seconds:

- ❑ There can be only so many different types of performance requirements (~10)
- ❑ Only so many inter-connected DCs (a few 10s)

Rate of Convergence



Three-tier: Number of iterations to converge for different combinations of class-level and DC-level step sizes.

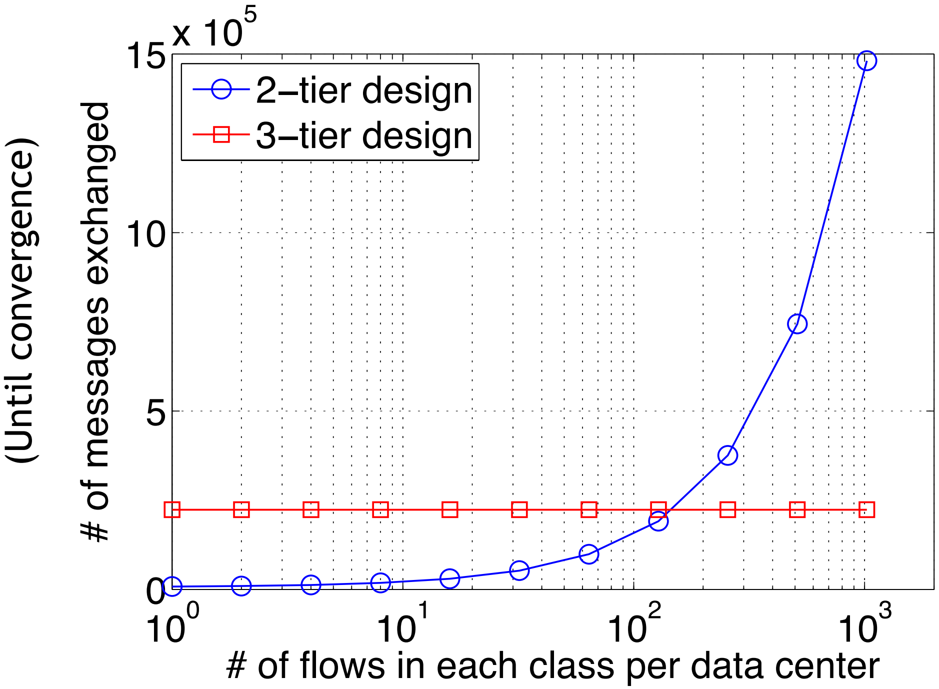
Rate of Convergence

Summary of the convergence behavior

Class-level step size β	2-tier design	3-tier design
small $\beta = 1, 2$	slow	very slow, all α
medium $\beta = 5, 10$	moderate	slow, all α
large $\beta = 20, 30$	fast	moderate, all α
very large $30 < \beta < 40$	fast	moderate, $\alpha \leq 16$
extremely large $40 \geq \beta < 50$	fast	does not converge
$\beta \geq 50$	does not converge	does not converge

- In practice, choose step sizes that converge quickly.
- Dynamic traffic demand: For private OSP backbone, the demand variability can be controlled to some extent

Message-Passing Overhead



K	No. of classes
L	No. of backbone links
J	No. of DCs
N N' 100	No. of class-level allocations to converge in two-tier and three-tier designs
M 5	No. of DC-level allocations to converge in three-tier

- Messages are sent over the wide area network
- Number of messages depends on the number of flows in the two-tier design, but not in the three-tier design
- Small compared to the total traffic volume

Two-tier:
# of variables

$$N\left(2KL + \sum_k \sum_j \sum_{s\in\mathcal{F}^{kj}} \sum_p R^k_{sp}\right)$$

Three-tier:
# of variables

$$N'\left(2KL + 2JKLM\right)$$

Conclusions

- Software defined traffic management for wide area data center backbone networks
- Two scalable and practical semi-centralized designs using a small number of controllers that can be implemented in real-world data center backbones (Google)
- Joint rate control, routing, and link scheduling using optimization in a modular, tiered design
- Results provably optimal using principles of optimization decomposition
- Tradeoff between rate of convergence and message-passing – choose the design that suits the OSP best

Thank You

- Amitabha Ghosh, Sangtae Ha, Edward Crabbe, and Jennifer Rexford, "*Scalable Multi-Class Traffic Management in Data Center Backbone Networks*," IEEE JSAC: Networking Challenges in Cloud Computing Systems and Applications, vol. 13, no. 12, 2013 (in press).

http://anrg.usc.edu/~amitabhg/papers/JSAC-CloudComputing-2013.pdf

Puzzles

Please take a look at the following links:

1. http://gurmeet.net/puzzles/
2. http://www.dcg.ethz.ch/members/roger/puzzles/
3. http://research.microsoft.com/en-us/um/people/leino/puzzles.html
4. (Lateral Thinking Puzzles) http://www.thecourse.us/students/lateral_thinking.htm

Engineers and Salary

- Four honest and hard-working computer engineers are sipping coffee at Starbucks. They wish to compute their average salary. However nobody is willing to reveal an iota of information about his/her own salary to anybody else.

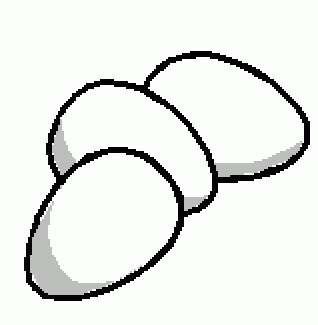
- Question: Is it possible? If so, how do they do it?

Doors

- There are 100 doors in a row that are all initially closed. You make 100 passes by the doors starting with the first door every time.
  - First time you visit every door and toggle the door (if the door is closed, you open it, if its open, you close it).
  - Second time you only visit every 2nd door (door #2, #4, #6), and toggle.
  - Third time, every 3rd door (door #3, #6, #9), etc., until you only visit the 100th door.

- Question: What state (open / closed) are the doors in after the 100th pass?

Egg Drop



- You have two identical eggs. You can access a 100-story building.
- You are told that if you drop an egg from or above a particular floor the egg will break.

- Question:
  - You need to figure out the highest floor an egg can be dropped without breaking.
  - How many drops you need to make? You are allowed to break both the eggs in the process.

Blind Man and Cards

- A blind man is handled a deck of 52 cards, and told that exactly 10 of these cards are facing up.

- Question: How can he divide the cards into two piles, not necessarily of equal size, with each pile having the same number of cards facing up?

Baskets, Apples, and Oranges

- Basket 1 has two apples
- Basket 2 has two oranges
- Basket 3 has one apple and one orange
- Each basket has a label “Apple”, “Orange”, or “Apple & Orange” – But all the labels are wrong!
- You are allowed to open one basket, pick one fruit, see it, and put it back into the basket (you don’t get to see the other fruit)

- Question: How many such operations are necessary to correctly label the baskets?

Cap Colors

- An evil troll once captured a bunch of gnomes and told them:
  - "Tomorrow, I will make you stand in a file, ordered by height such that a gnome can see exactly those gnomes that are shorter than him.”
  - “I will place a cap (one of 10 different colors) on each head.”
  - “Then, starting from the tallest, each gnome has to declare aloud what he thinks the color of his own cap is.”
  - “In the end, those who were correct will be spared; others will be eaten, silently."
- The gnomes set thinking and came up with a strategy.

- Question: How many of them survived?