Scalable Multi-Class Traffic Management in Data Center Backbone Networks

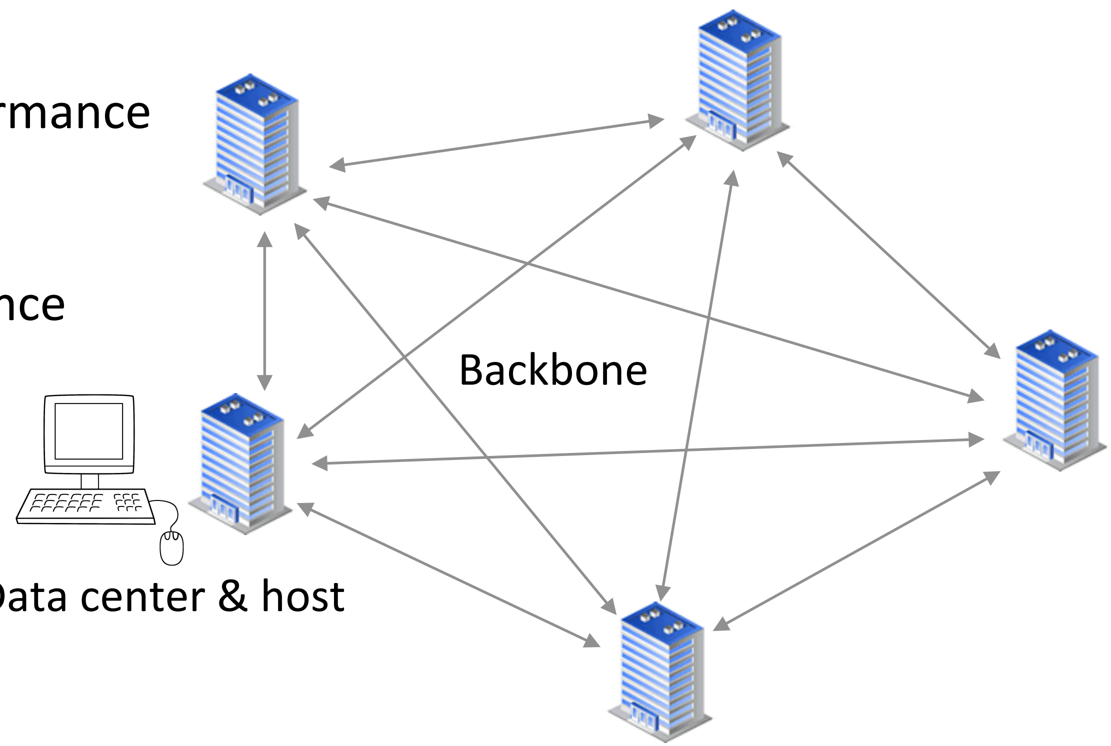
Amitabha Ghosh (UtopiaCompression)
Sangtae Ha (Princeton)
Edward Crabbe (Google)
Jennifer Rexford (Princeton)

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

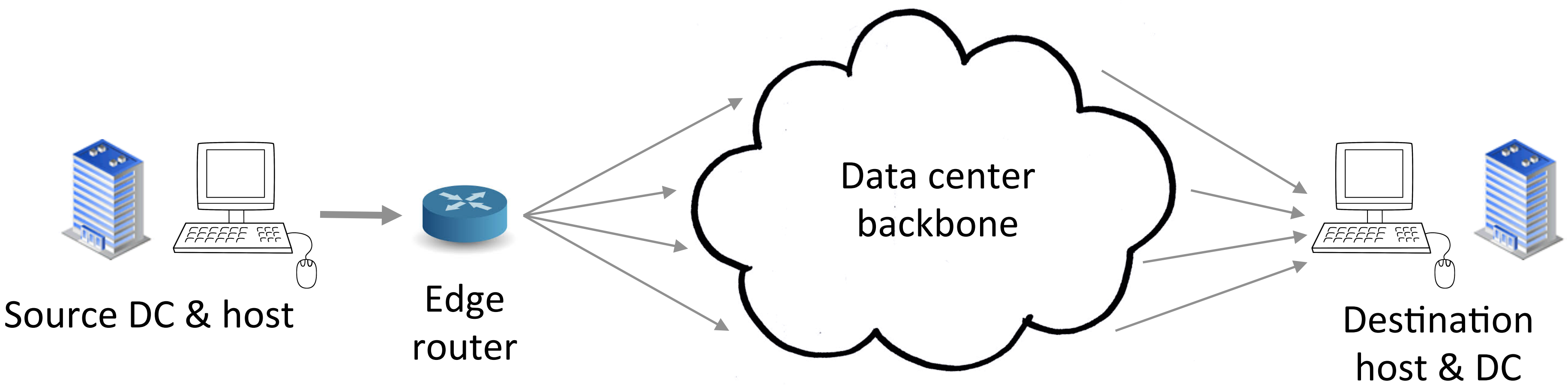


Contributions

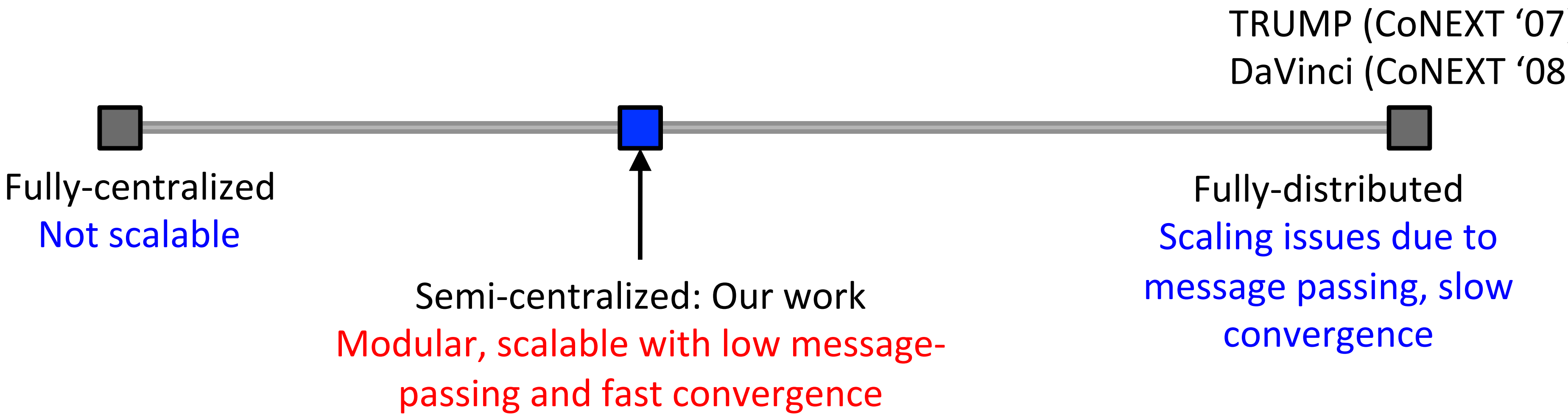
Controlling the three "knobs"

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

Joint optimization of rate control, routing, and link scheduling



Contributions



- ❑ 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

  $\mathbf{A} = [A_{lp}]$ Topology matrix smaller

  $\mathbf{R} = [R_{sp}^k]$ Path routing matrix larger
- ❑ One queue per class
- ❑ Multi-path routing
  - Path rate of flow s of class k over path p z_{sp}^k

Model and Formulation

Utility of Flow s of Class k

Coefficients to model different degrees of sensitivity to throughput and delay

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

- w_s^k: Weight of flow s of class k
- f^k: Throughput sensitivity of class k, e.g., log(.)
- x_s^k: Total sending rate of flow s of class k
- g^k: Delay sensitivity of class k → Sum of the products of path rates and average end-to-end delays on those paths
- u_l^k: Utilization of class k over link l

Model and Formulation

Objective Function

- ❑ Data centers, backbone and traffic sources under the same OSP ownership

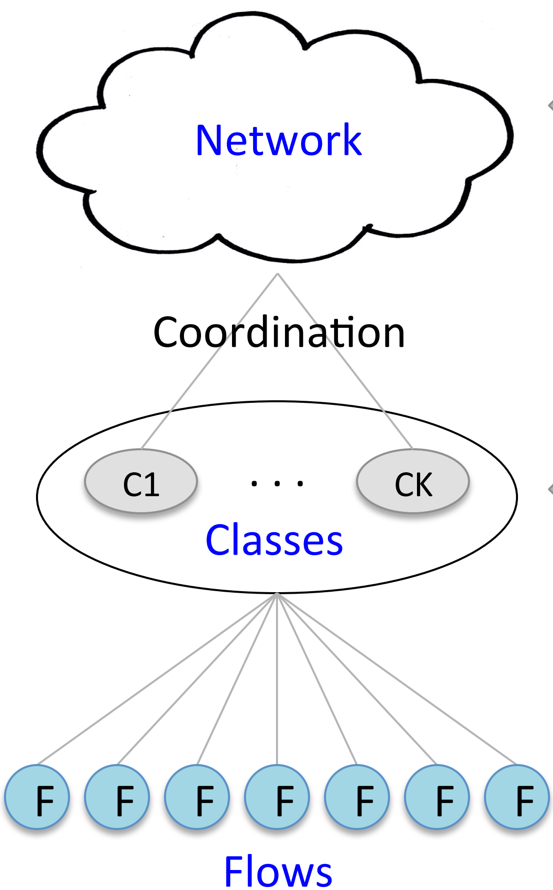
- ❑ Maximize the sum of utilities of all flows across all traffic classes (global "social welfare")

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

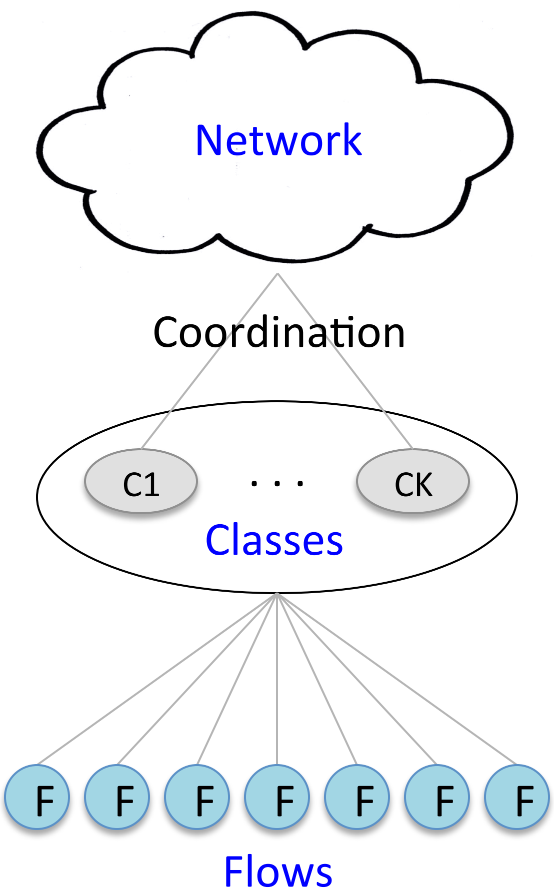
- A centralized entity
- Knows network topology

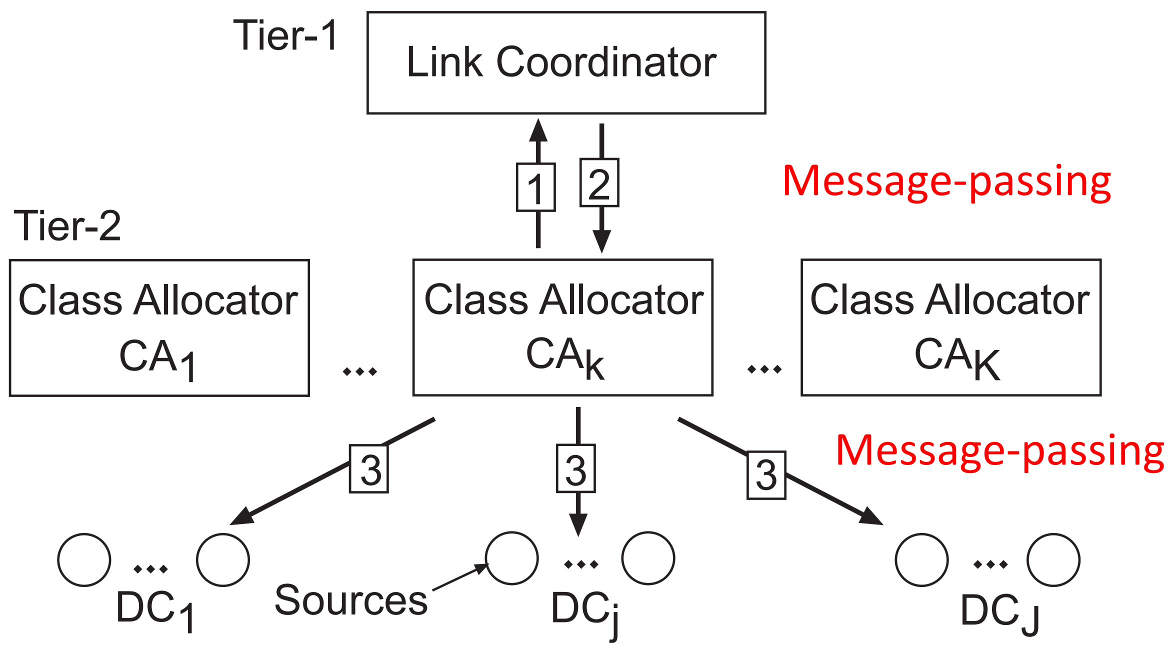
$\mathbf{z}^k$

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

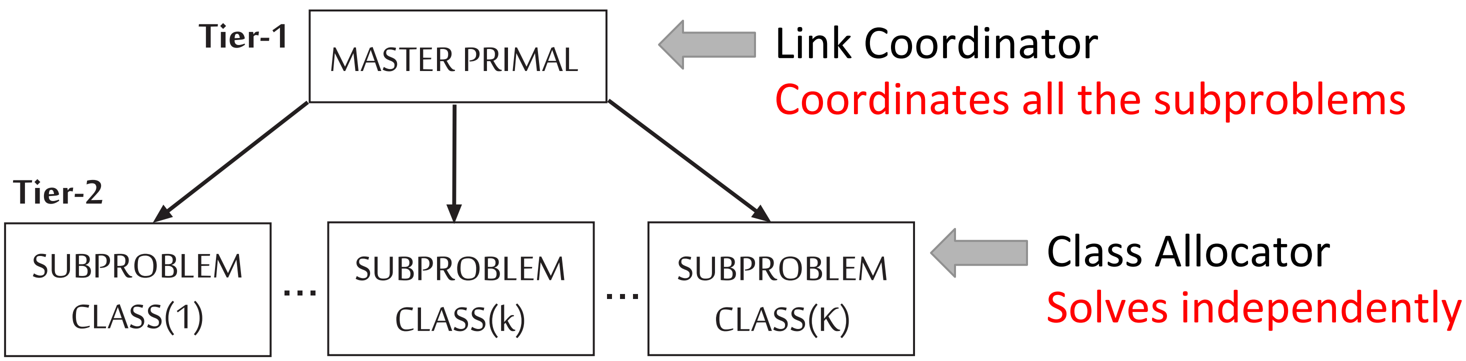
Two-Tier Design





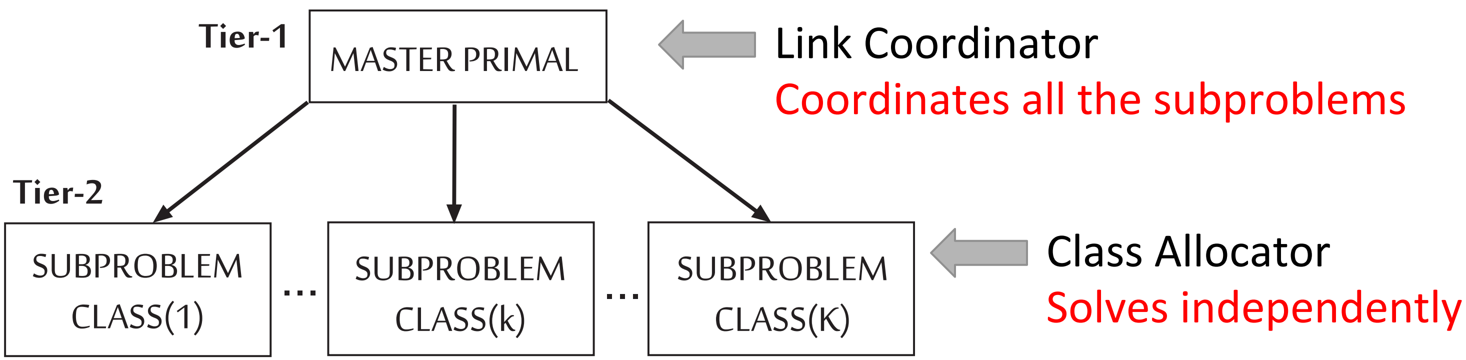
Two-Tier Decomposition

Primal Decomposition



Two-Tier Decomposition

Primal Decomposition

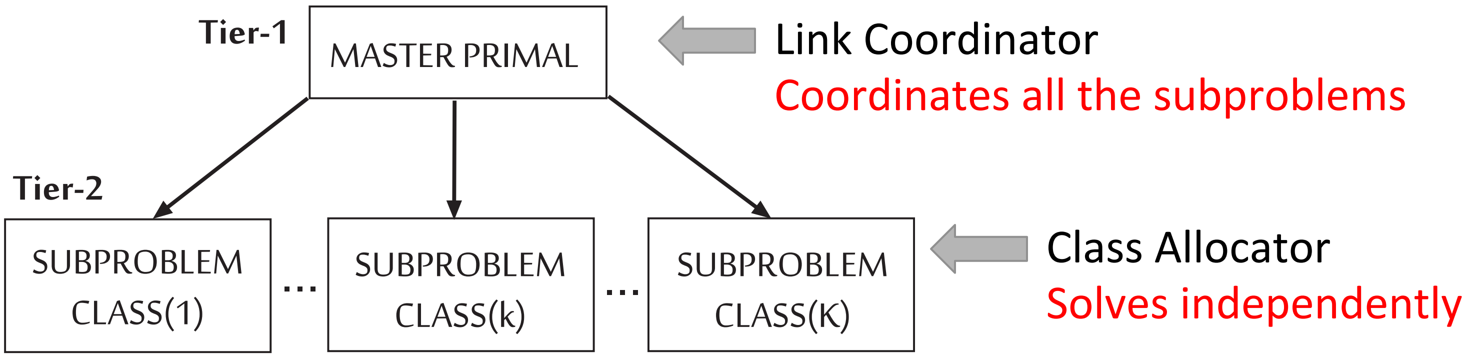


Subproblem for Class k

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

Two-Tier Decomposition

Primal Decomposition



Subproblem for Class k

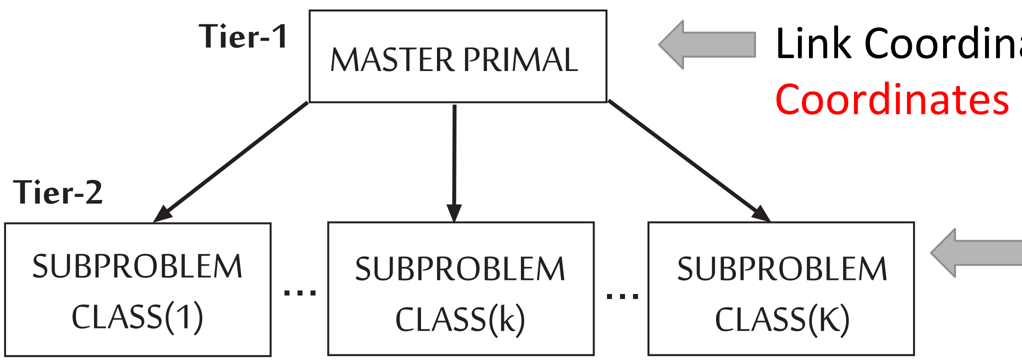
$$\begin{aligned} \text{maximize} \quad & \mathcal{U}^k = \sum_{s \in \mathcal{F}^k} U_s^k \\ \text{subject to} \quad & \mathbf{A}\mathbf{R}^k \mathbf{z}^k \preceq \mathbf{y}^k \quad \forall k \\ \text{variables} \quad & \mathbf{z}^k \succeq 0 \end{aligned}$$

Master Primal

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

Two-Tier Decomposition

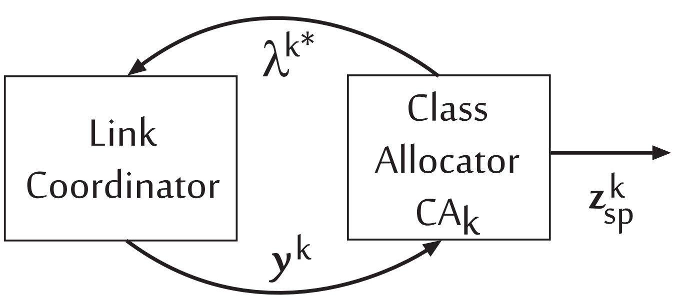
Primal Decomposition



Link Coordinator
Coordinates all the subproblems

Class Allocator
Solves independently

Message-Passing

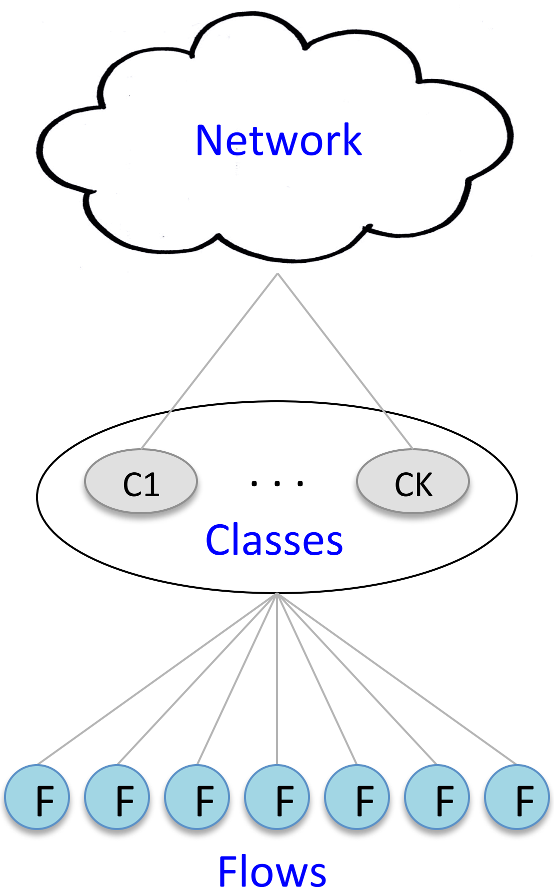


step size : β

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

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

Three-Tier Design



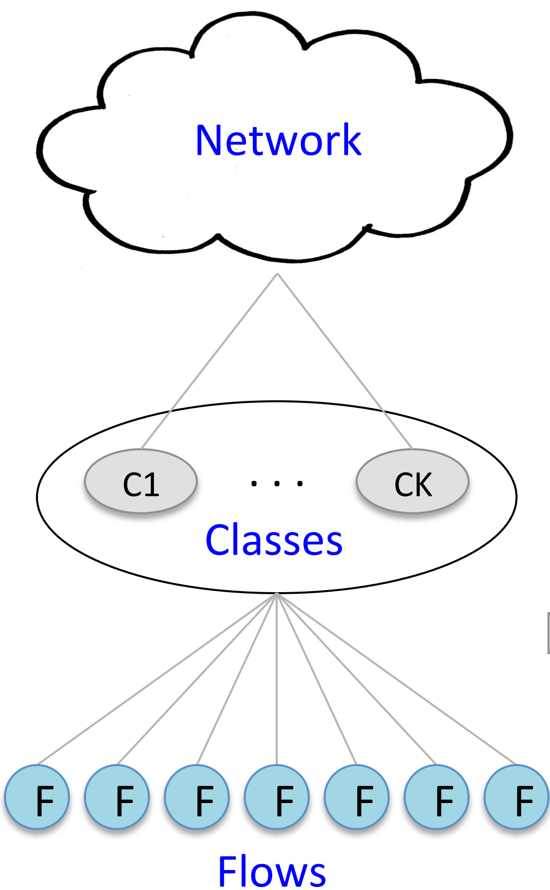
Why another tier? (High control overhead)

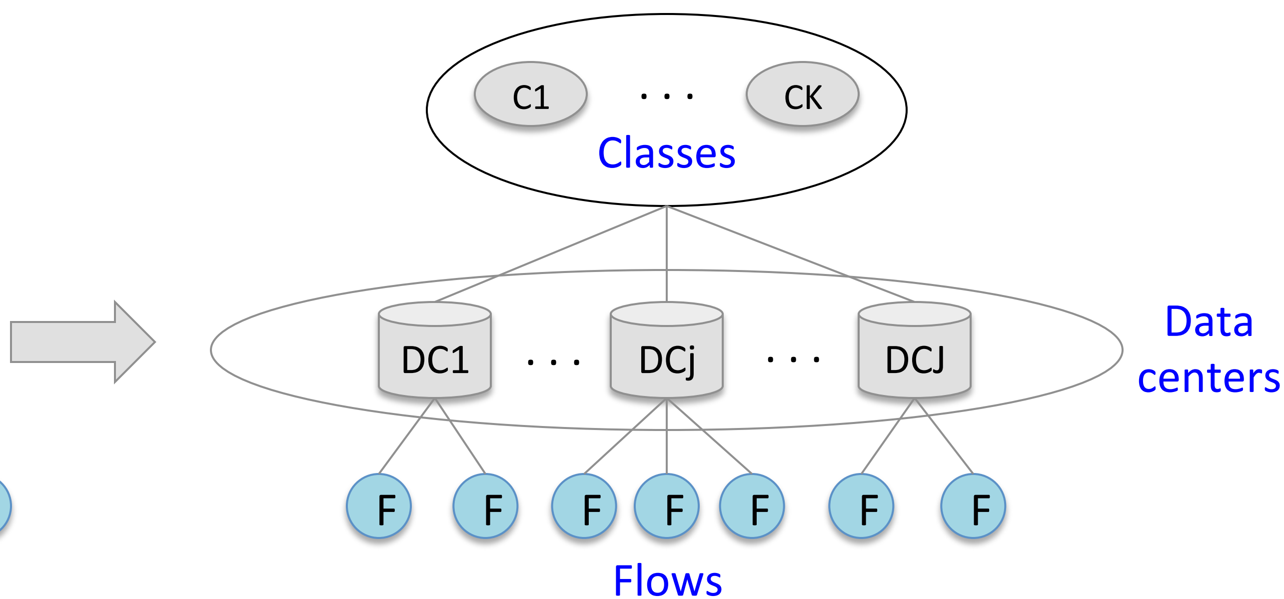
- Flow of a given class may originate from any DC
- Each class allocator potentially communicates with all DCs

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

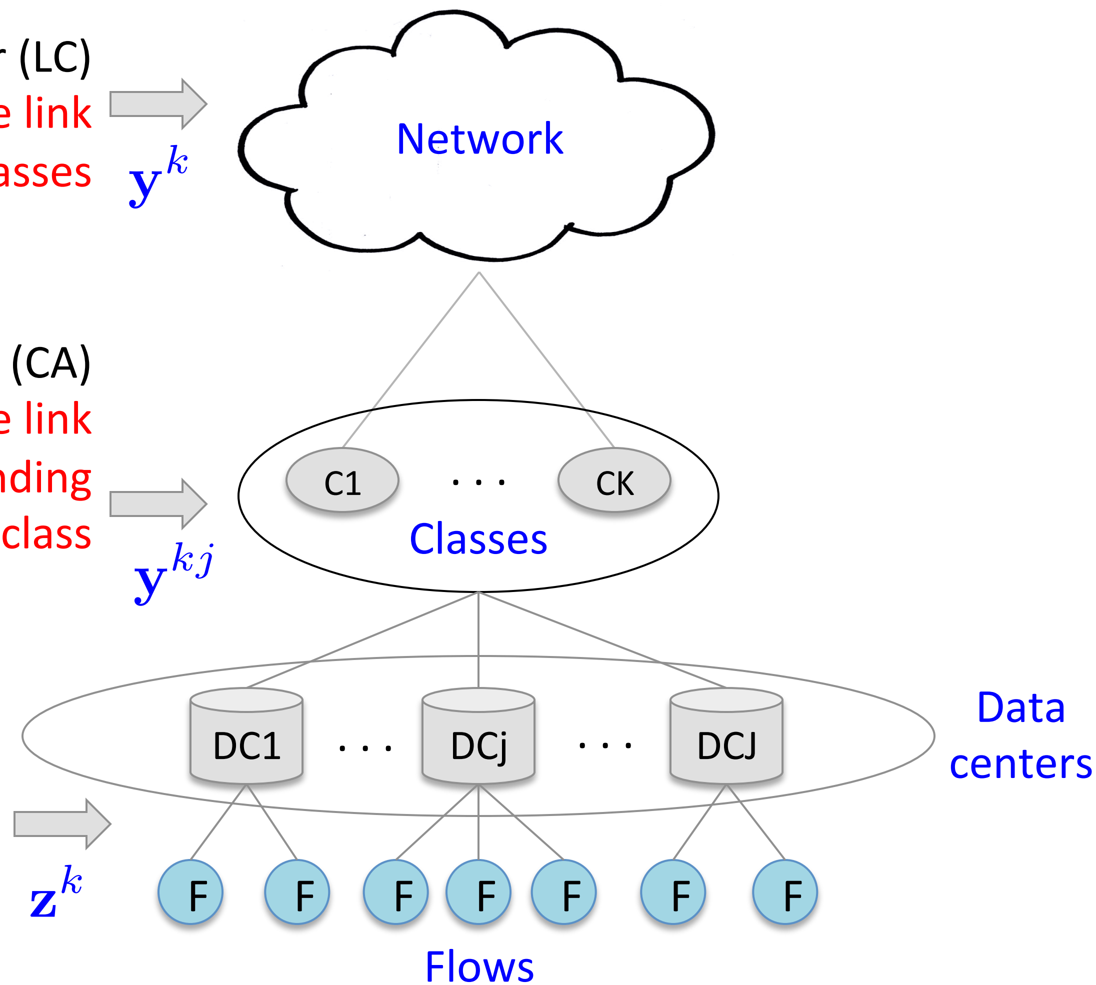
Link Coordinator (LC)
Optimizes aggregate link bandwidth across classes

One per class

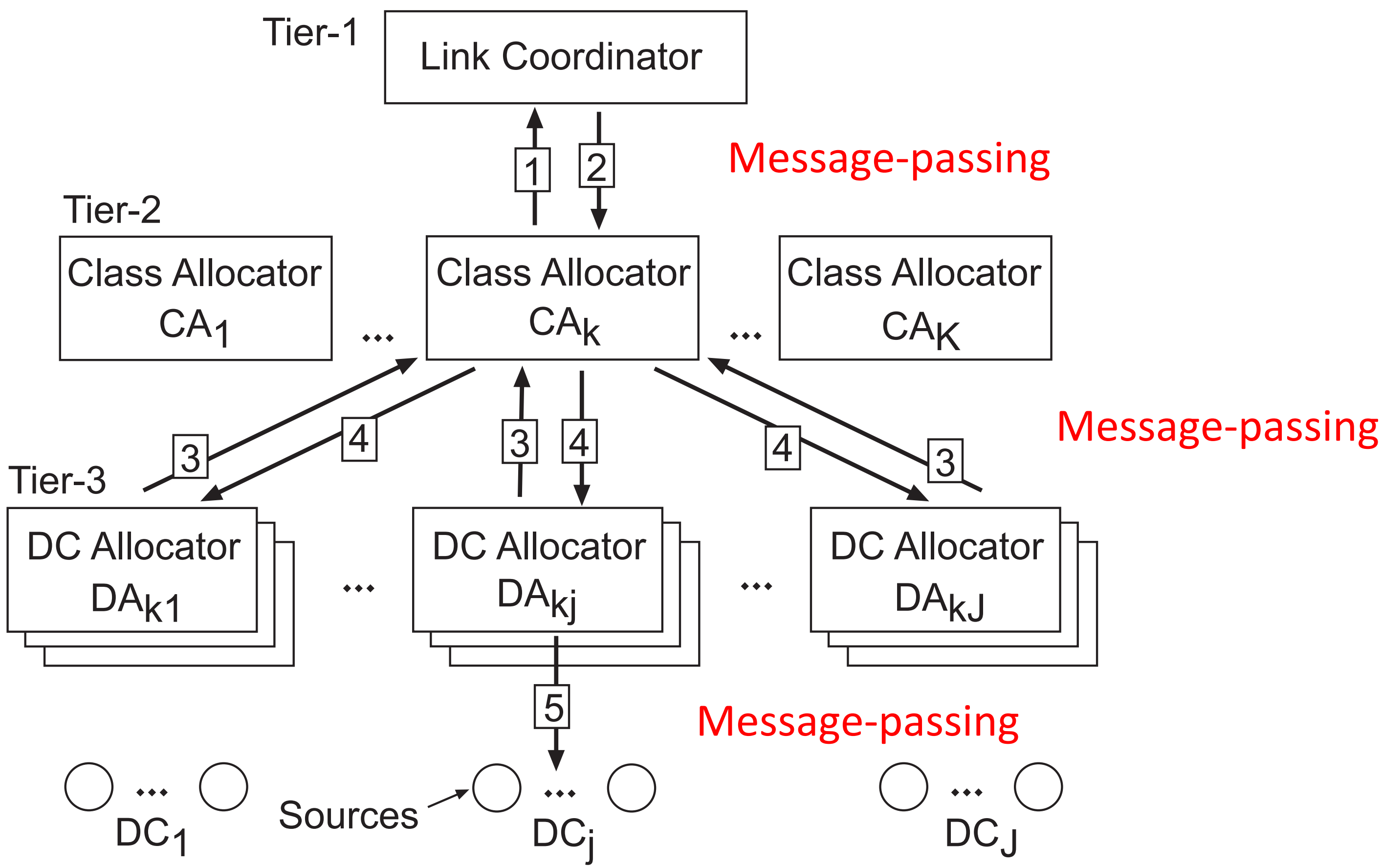
Class Allocator (CA)
Optimizes aggregate link bandwidth across DCs sending traffic in its own class

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

One per class, per DC

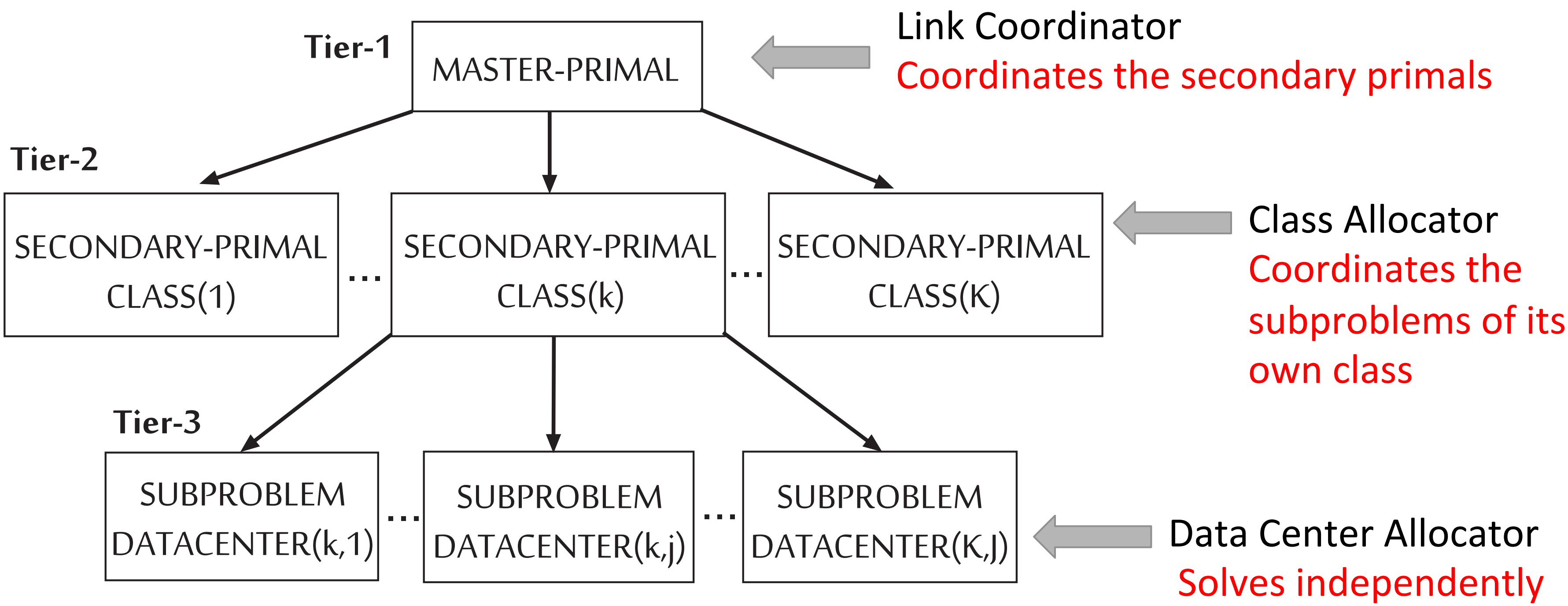


Three-Tier Design



Three-Tier Decomposition

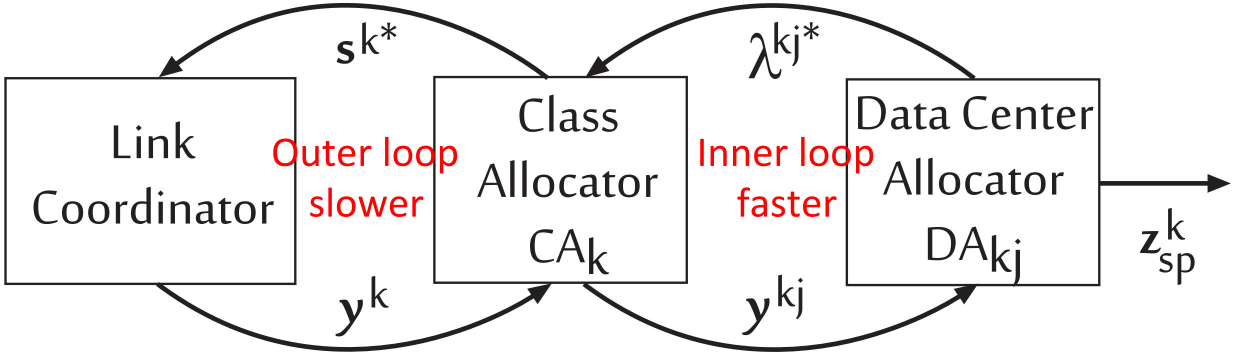
2-Level Primal Decomposition



Three-Tier Decomposition

Message-Passing

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



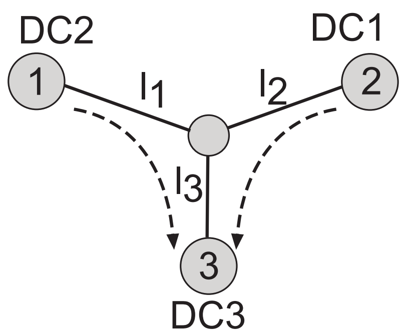
$\mathbf{y}^{k}$: Aggregate bandwidth assigned to class k

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

Performance Evaluation

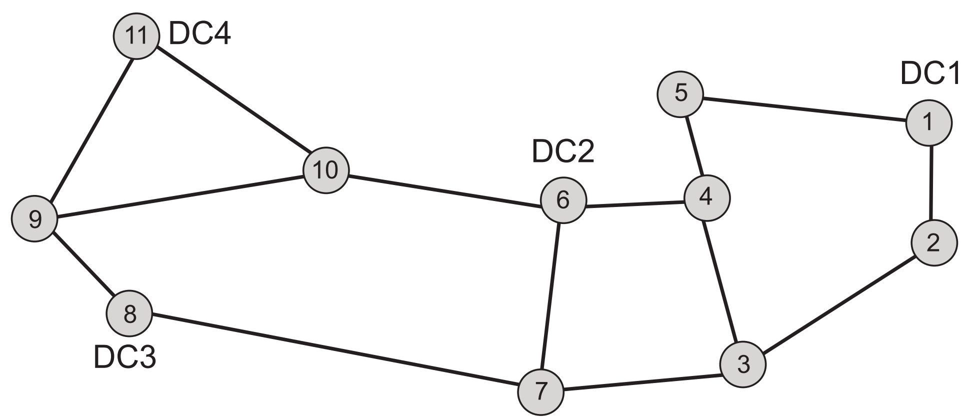
Performance Metrics

- ❑ Rate of convergence
- ❑ Message-passing overhead



Simple topology

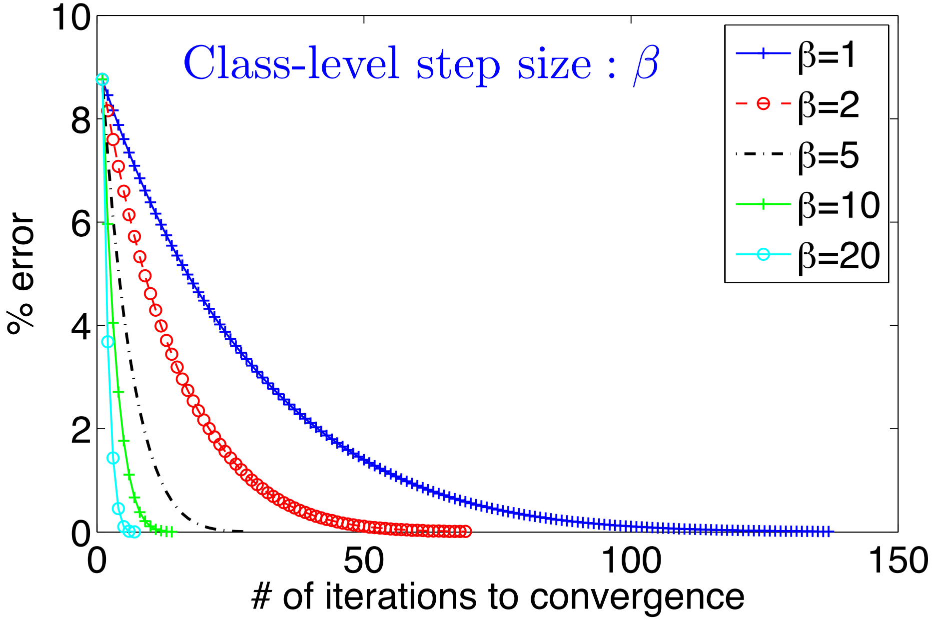
DC1 & DC2 send traffic to DC3
100 Mbps link capacity
Two classes with log utility



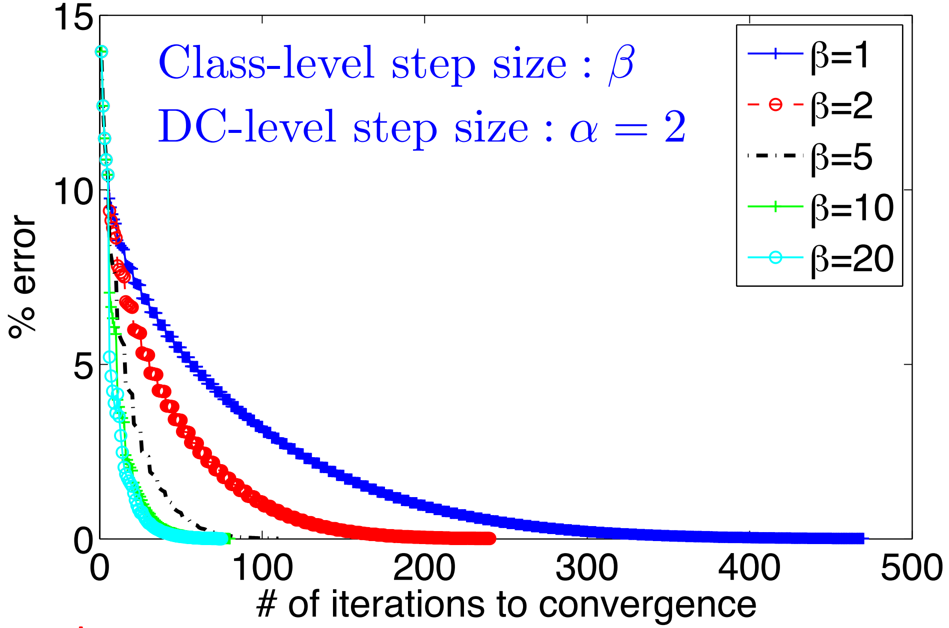
Abilene topology

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

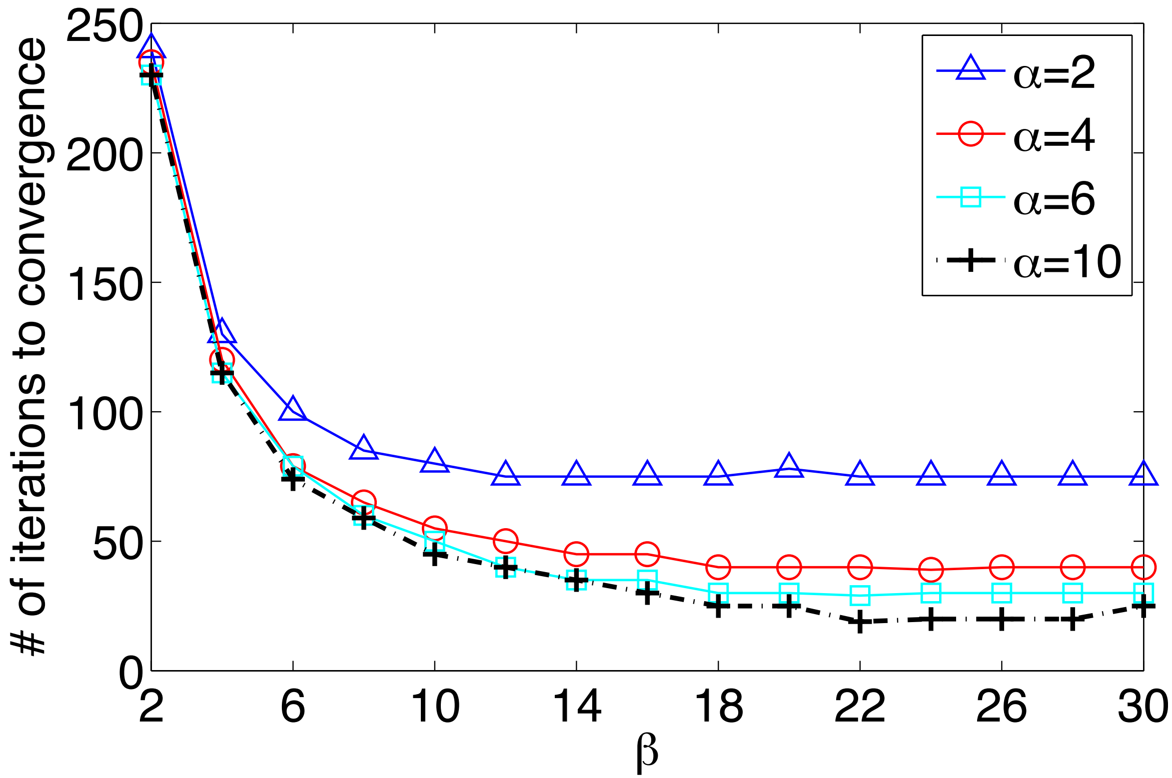


Three-tier

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.

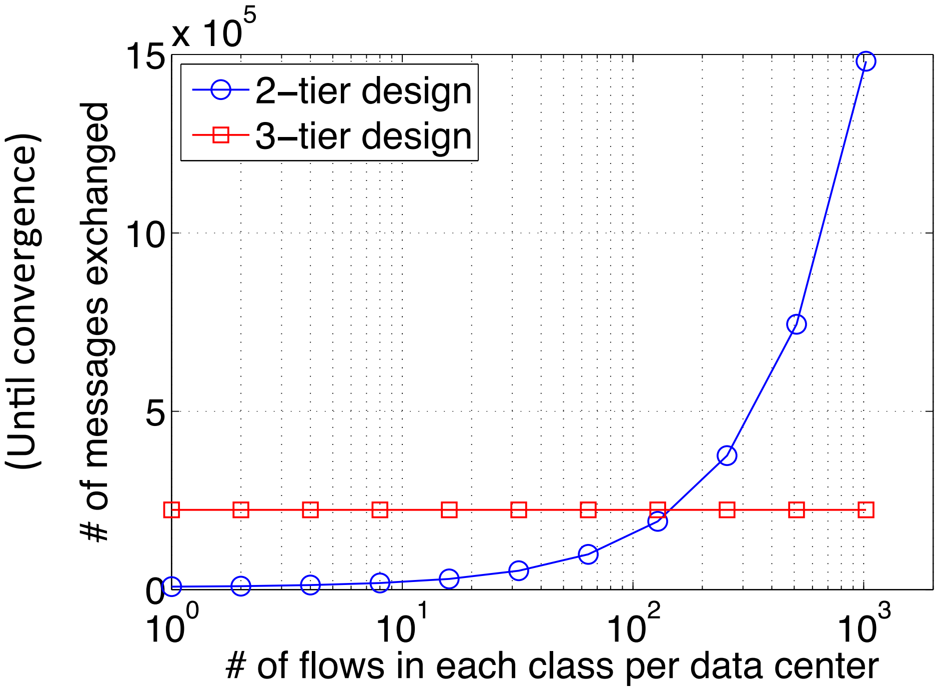
Rate of Convergence

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

$$N\left(2KL + \sum_{k}\sum_{j}\sum_{s\in\mathcal{F}^{kj}}\sum_{p} R_{sp}^{k}\right)$$

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