FairCloud

Presented by
Aishwarya Parasuram

Problem Statement

To create a mechanism of cloud sharing that:

- provides minimum bandwidth guarantees
- allocates the network proportional to payment

Why is sharing the cloud service difficult?

The network allocation of a VM (x) depends on:

- other VMs running on the same machine with x
- other VMs that x communicates with
- cross-traffic on each link used by x

Currently:

- Cloud services are shared in a best-effort manner
- Neither tenants nor cloud-providers can reason about how network resources are allocated

Goals of the Paper

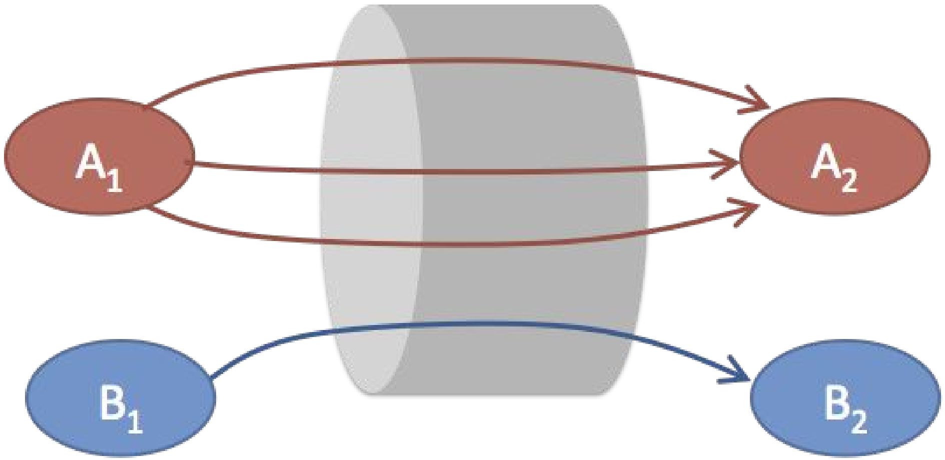
1. Propose a set of desirable properties for allocating network bandwidth at VM granularity
2. Expose fundamental trade-offs in network resource allocation
3. Show that existing policies violate one or more of the above properties
4. Propose a mechanism that can achieve a large subset of desirable properties, and tries to overcome trade-offs

Basic Assumptions

- IaaS model (eg: Amazon EC2)
  - Tenants pay fixed flat-rate per VM
  - Goals for network sharing are defined from a per-VM point of view
- All VMs are identical and have the same price
- Discussion is agnostic to VM placement and routing
- Orthogonal to work on network topologies aimed at improvising bisection bandwidth
  - possibility of congestion (and thereby the need for sharing policies) remains even in full bisection-bandwidth networks
  - Eg: Many-to many link in MapReduce can congest any of the links in the networks

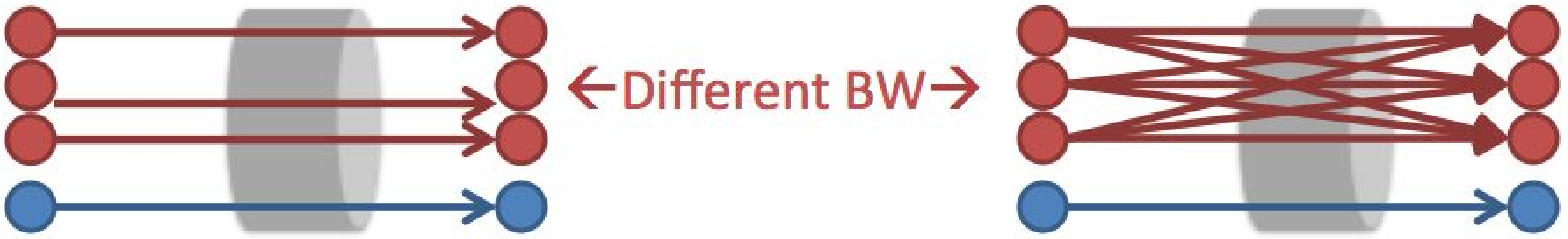
Traditional Allocation Policies

- **Per-flow mechanism**
  - S-D pair can initiate more flows to get more BW



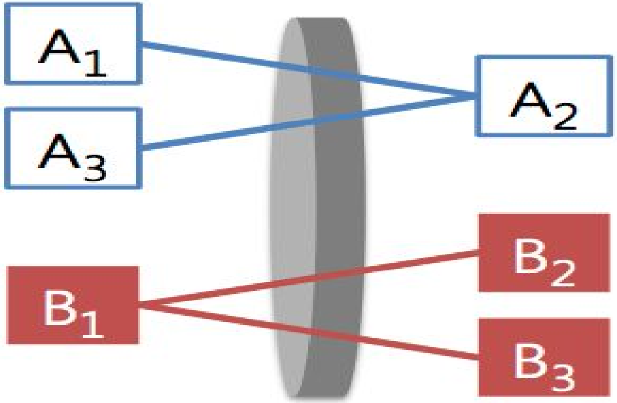
Traditional Allocation Policies

- Per-flow mechanism
- **Source-Destination pair**
  - Many-to-many gets more BW than one-to-one



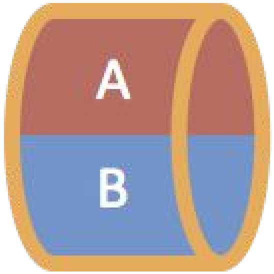
Traditional Allocation Policies

- Per-flow mechanism
- Source-Destination pair
- **Per-source / Per-Destination (Seawall, NSDI '11)**
  - Asymmetric - application level inefficiencies
  - No link proportionality or min-guarantee

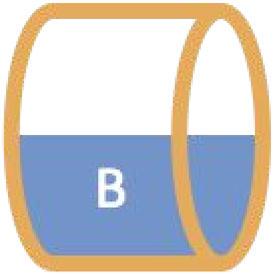


Traditional Allocation Policies

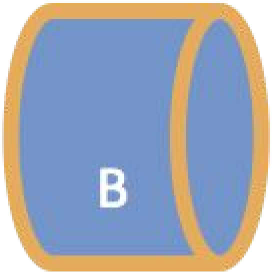
- Per-flow mechanism
- Source-Destination pair
- Per-source / Per-Destination (Seawall, NSDI ‘11)
- **Static Allocation (Oktopus, Sigcomm ‘12)**



Congested N/W



A Stops sending

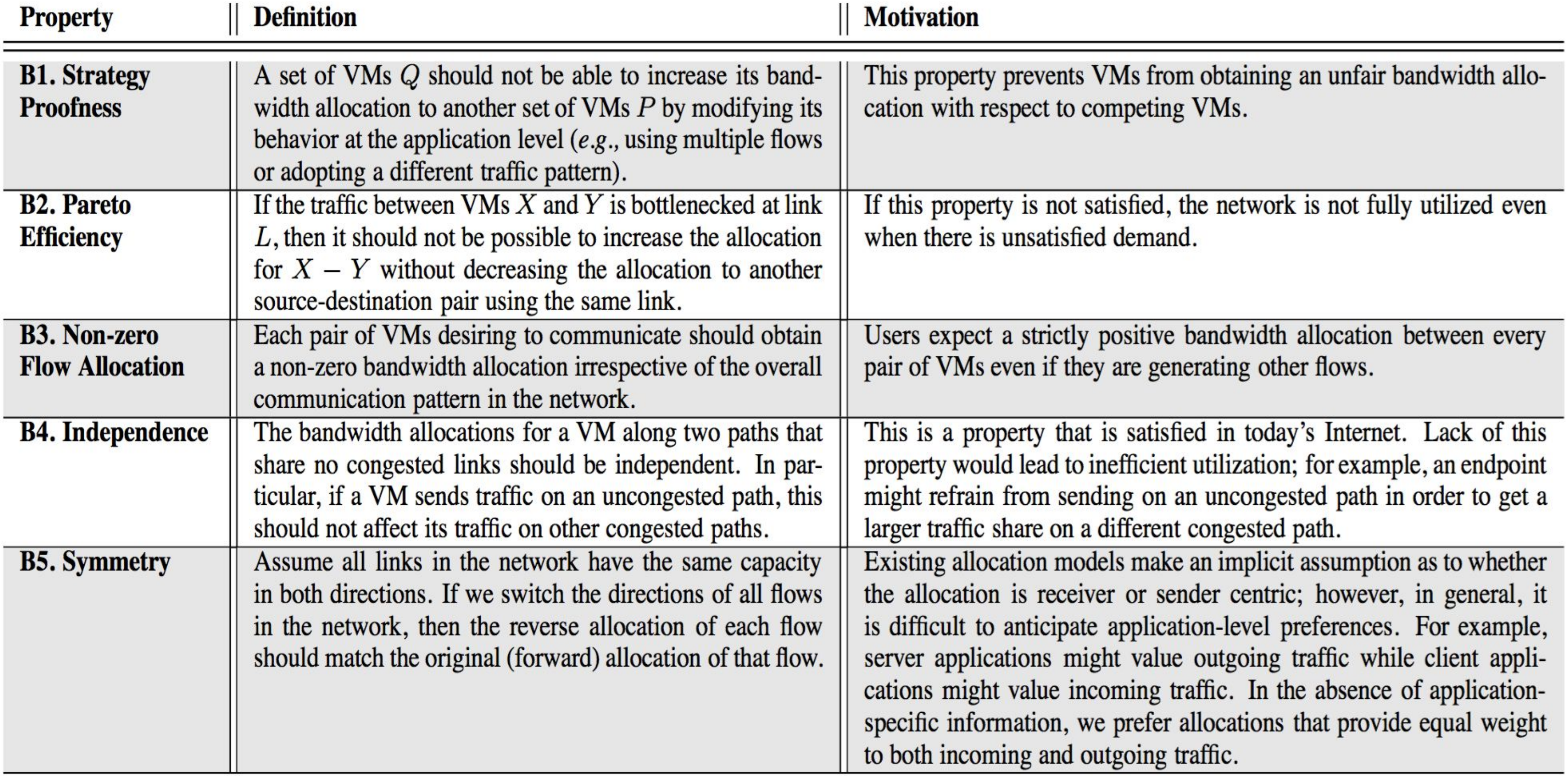


Ideal

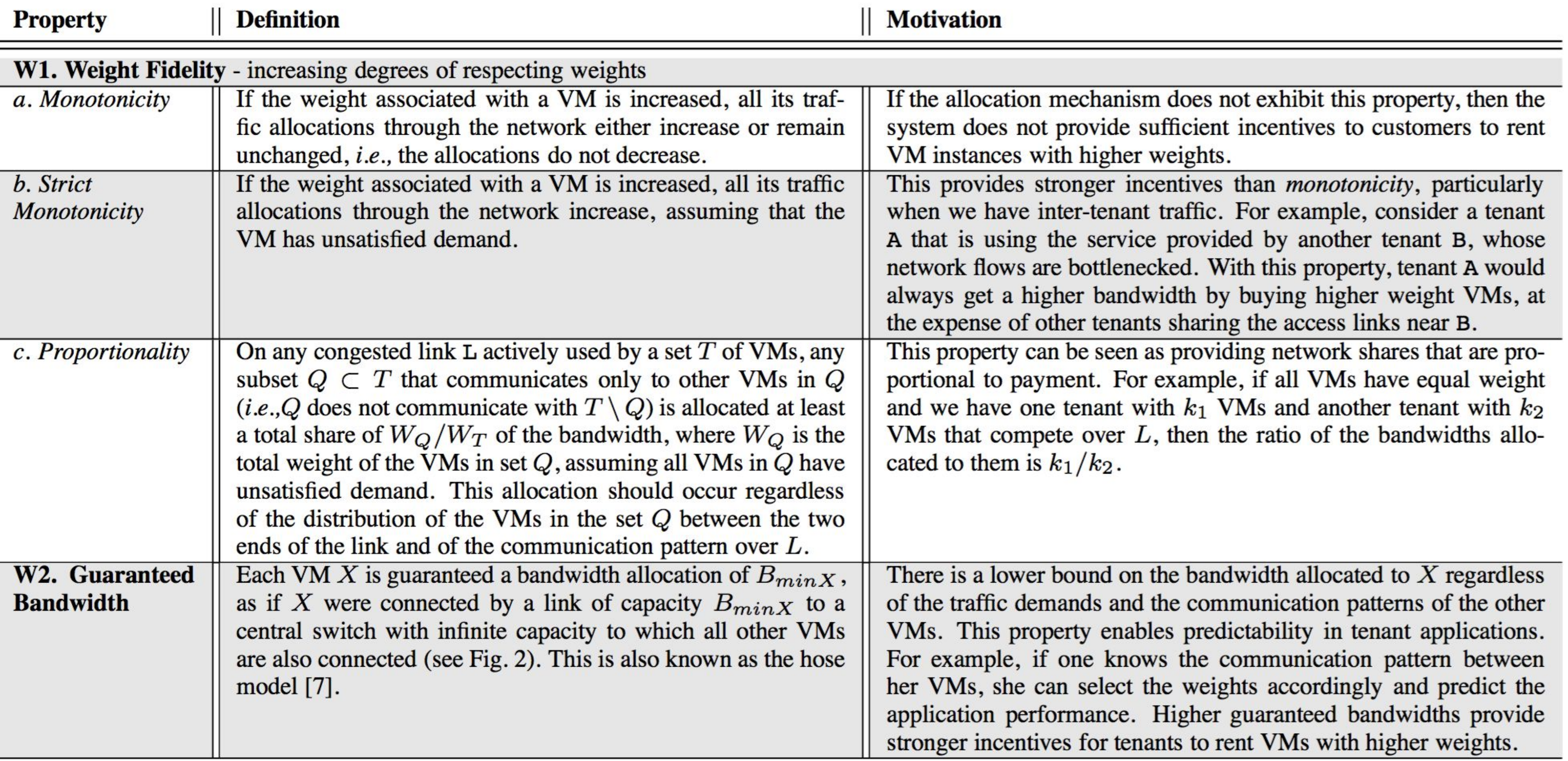
Key Ideas

1. Allocate BW along congested links in proportion to the source and destination weights
   a. not to the number of flows, sources or source-destination pairs of the tenant
2. Use VM’s proximity to a link to compute tenant’s share on that link
   a. The share of a tenant on a link in a tree-based topology is computed as a function of the sum of VMs in the tenant’s sub-tree

Desirable Properties



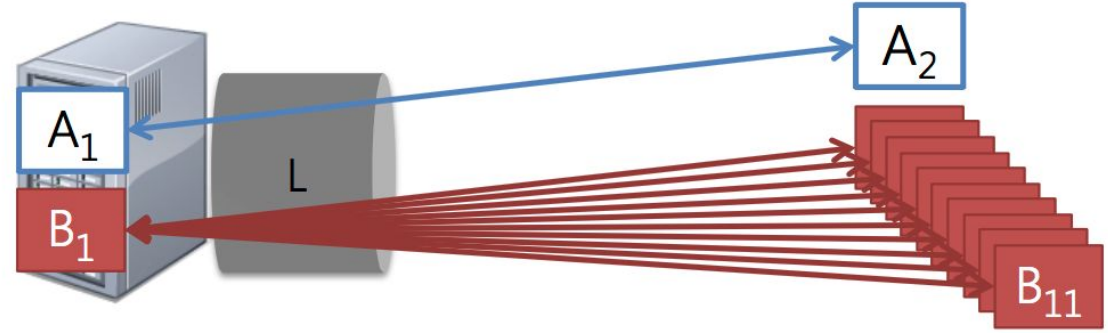
Property	Definition	Motivation
B1. Strategy Proofness	A set of VMs Q should not be able to increase its bandwidth allocation to another set of VMs P by modifying its behavior at the application level (*e.g.*, using multiple flows or adopting a different traffic pattern).	This property prevents VMs from obtaining an unfair bandwidth allocation with respect to competing VMs.
B2. Pareto Efficiency	If the traffic between VMs X and Y is bottlenecked at link L, then it should not be possible to increase the allocation for $X - Y$ without decreasing the allocation to another source-destination pair using the same link.	If this property is not satisfied, the network is not fully utilized even when there is unsatisfied demand.
B3. Non-zero Flow Allocation	Each pair of VMs desiring to communicate should obtain a non-zero bandwidth allocation irrespective of the overall communication pattern in the network.	Users expect a strictly positive bandwidth allocation between every pair of VMs even if they are generating other flows.
B4. Independence	The bandwidth allocations for a VM along two paths that share no congested links should be independent. In particular, if a VM sends traffic on an uncongested path, this should not affect its traffic on other congested paths.	This is a property that is satisfied in today's Internet. Lack of this property would lead to inefficient utilization; for example, an endpoint might refrain from sending on an uncongested path in order to get a larger traffic share on a different congested path.
B5. Symmetry	Assume all links in the network have the same capacity in both directions. If we switch the directions of all flows in the network, then the reverse allocation of each flow should match the original (forward) allocation of that flow.	Existing allocation models make an implicit assumption as to whether the allocation is receiver or sender centric; however, in general, it is difficult to anticipate application-level preferences. For example, server applications might value outgoing traffic while client applications might value incoming traffic. In the absence of application-specific information, we prefer allocations that provide equal weight to both incoming and outgoing traffic.



Property	Definition	Motivation
W1. Weight Fidelity - increasing degrees of respecting weights		
a. Monotonicity	If the weight associated with a VM is increased, all its traffic allocations through the network either increase or remain unchanged, *i.e.,* the allocations do not decrease.	If the allocation mechanism does not exhibit this property, then the system does not provide sufficient incentives to customers to rent VM instances with higher weights.
b. Strict Monotonicity	If the weight associated with a VM is increased, all its traffic allocations through the network increase, assuming that the VM has unsatisfied demand.	This provides stronger incentives than *monotonicity*, particularly when we have inter-tenant traffic. For example, consider a tenant A that is using the service provided by another tenant B, whose network flows are bottlenecked. With this property, tenant A would always get a higher bandwidth by buying higher weight VMs, at the expense of other tenants sharing the access links near B.
c. Proportionality	On any congested link L actively used by a set T of VMs, any subset $Q \subset T$ that communicates only to other VMs in Q (*i.e.,*Q does not communicate with $T \setminus Q$) is allocated at least a total share of W_Q/W_T of the bandwidth, where W_Q is the total weight of the VMs in set Q, assuming all VMs in Q have unsatisfied demand. This allocation should occur regardless of the distribution of the VMs in the set Q between the two ends of the link and of the communication pattern over L.	This property can be seen as providing network shares that are proportional to payment. For example, if all VMs have equal weight and we have one tenant with k_1 VMs and another tenant with k_2 VMs that compete over L, then the ratio of the bandwidths allocated to them is k_1/k_2.
W2. Guaranteed Bandwidth	Each VM X is guaranteed a bandwidth allocation of B_{minX}, as if X were connected by a link of capacity B_{minX} to a central switch with infinite capacity to which all other VMs are also connected (see Fig. 2). This is also known as the hose model [7].	There is a lower bound on the bandwidth allocated to X regardless of the traffic demands and the communication patterns of the other VMs. This property enables predictability in tenant applications. For example, if one knows the communication pattern between her VMs, she can select the weights accordingly and predict the application performance. Higher guaranteed bandwidths provide stronger incentives for tenants to rent VMs with higher weights.

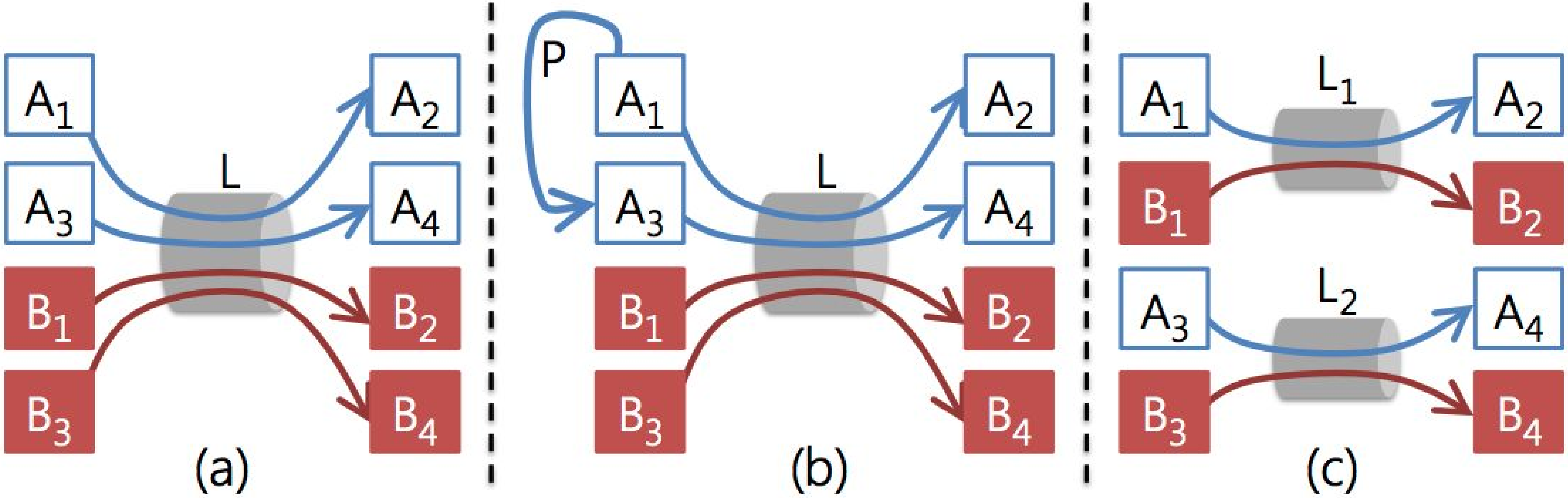
Trade-Offs

1. **Guaranteed BW and weight-fidelity (hard trade-off)**



Trade-Offs

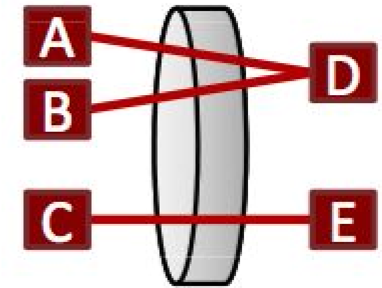
1. Guaranteed BW and weight-fidelity (hard trade-off)
2. **Weight-fidelity and high utilization**



Proposed Allocation Policies

1. Per End-point Sharing (PES)
2. One-sided Per End-point Sharing (OSPES)

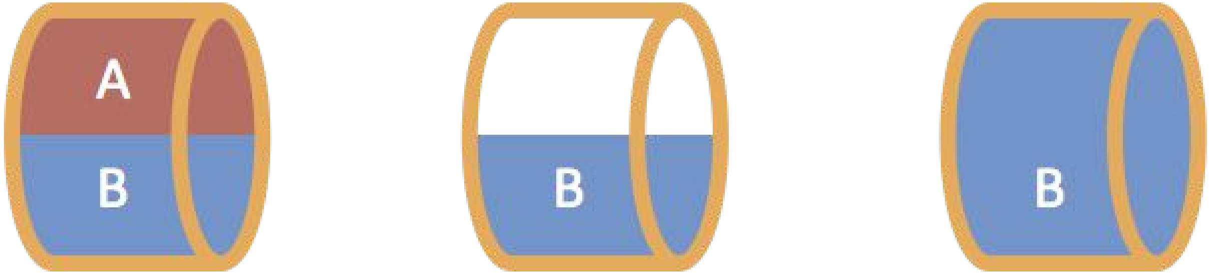
Per Endpoint Sharing (PES)



$$W_{A-B} = \frac{W_A}{N_A} + \frac{W_B}{N_B}$$

Per Endpoint Sharing (PES)

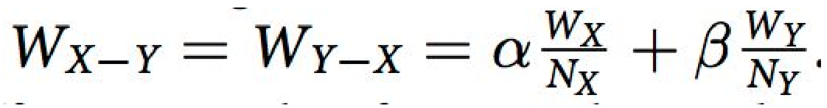
Drawback: Static Allocation, no work conservation



Also, guaranteed minimum BW is very low.

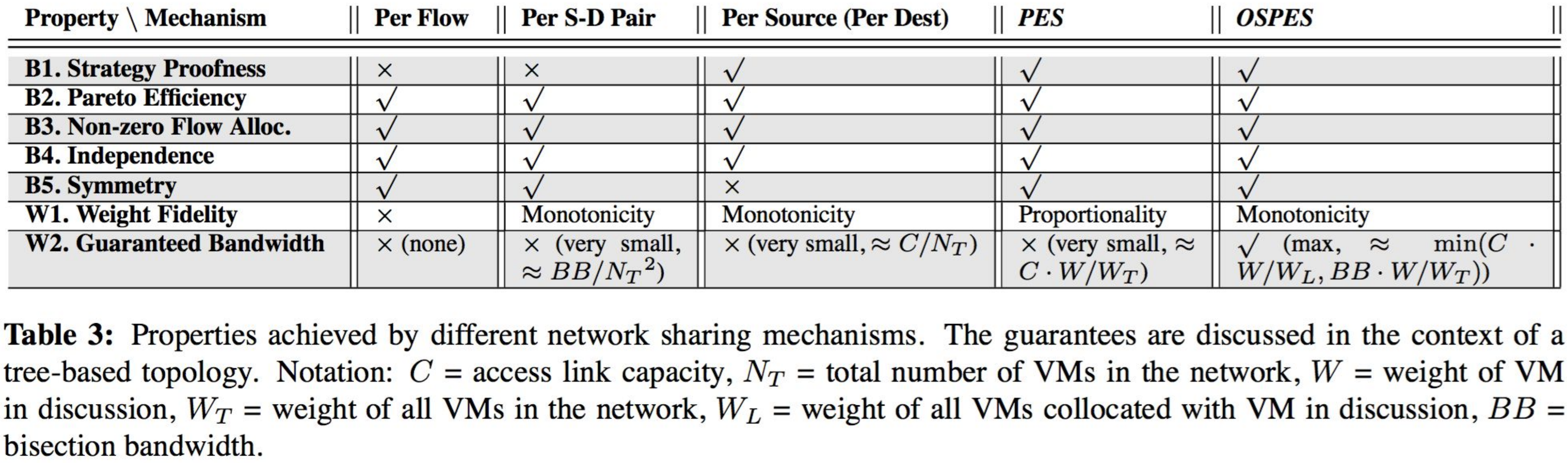
One-sided Per Endpoint Sharing (OSPES)

- Prioritizes VMs that are close to the link
- To offer higher worst-case BW guarantee:



$$W_{X-Y} = W_{Y-X} = \alpha\frac{W_X}{N_X} + \beta\frac{W_Y}{N_Y}.$$

- Optimized for tree-topologies: (1,0)
- Option: take demand into consideration while calculating weight for better work utilization



Property \ Mechanism	**Per Flow**	**Per S-D Pair**	**Per Source (Per Dest)**	***PES***	***OSPES***
B1. Strategy Proofness	×	×	√	√	√
B2. Pareto Efficiency	√	√	√	√	√
B3. Non-zero Flow Alloc.	√	√	√	√	√
B4. Independence	√	√	√	√	√
B5. Symmetry	√	√	×	√	√
W1. Weight Fidelity	×	Monotonicity	Monotonicity	Proportionality	Monotonicity
W2. Guaranteed Bandwidth	× (none)	× (very small, $\approx BB/{N_T}^2$)	× (very small, $\approx C/N_T$)	× (very small, $\approx C \cdot W/W_T$)	√ (max, $\approx \min(C \cdot W/W_L, BB \cdot W/W_T)$)

Table 3: Properties achieved by different network sharing mechanisms. The guarantees are discussed in the context of a tree-based topology. Notation: C = access link capacity, N_T = total number of VMs in the network, W = weight of VM in discussion, W_T = weight of all VMs in the network, W_L = weight of all VMs collocated with VM in discussion, BB = bisection bandwidth.

Main Findings from Deployment

- Trade-off between minimum BW and proportionality is also evident at network level
- Relative behaviors of these policies scale to large scale clusters

Discussions

1. How important are the desirable properties? Can they be ranked?
2. Are these the only “desirable” properties? Are there others?
   Eg: Destination based sharing prevents DoS attacks
3. Is there a mapping between user-requirements and properties? Formal way to express requirements in terms of properties
4. Is it really necessary to design a policy that achieves ALL of the desirable properties? Is it practically okay to make-do with existing ones?

The End