ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Performance of Network and Computing Resource Sharing in Federated Cloud Systems

Walter Cerroni
Dept. of Electrical, Electronic and Information Engineering
University of Bologna, Italy

walter.cerroni@unibo.it

Motivations

- Success of cloud platforms and services
  - significant savings in enterprise’s IT costs
  - increasing number of mobile cloud users (e.g., social media)

- Huge growth of cloud computing investments
  - public cloud market revenues in 2013: $ 58B
  - expected to reach $ 191B by 2020 (source: Forrester, 2014)

- Incresing demand of computing, storage and communication resources within Data Centers (DCs)
  - R&D on DC infrastructure technologies
  - advanced intra-DC and inter-DC networking solutions

Federated Cloud Computing

- DC over-provisioning may be too costly
  - expensive computing and communication equipment
  - energy consumption

- Distributed approach: Federated cloud systems
  - mutual agreement among different cloud providers
  - **workload shared** across multiple DC resources
  - increased **flexibility** and **mobility** of cloud services

- How to quantify the amount of computing and communication resources to be provided in the federation?
  - correctly **dimensioning the DC computing capacity** to be shared
  - efficiently **planning the underlying inter-DC network** infrastructure
  - **providing QoS**, considering the specific cloud service workload

Service Virtualization

- Service virtualization is widely used for DC administration and maintenance
  - **decoupling** service instances from underlying processing and storage hardware
  - key enabler for cloud federations

- Advantages of OS virtualization: Virtual Machines (VMs)
  - platform independency
  - quick deployment of new service instances
  - easy service replication and **migration** → flexibility and mobility
  - effective load balancing and server consolidation
  - easy backup and restore procedures

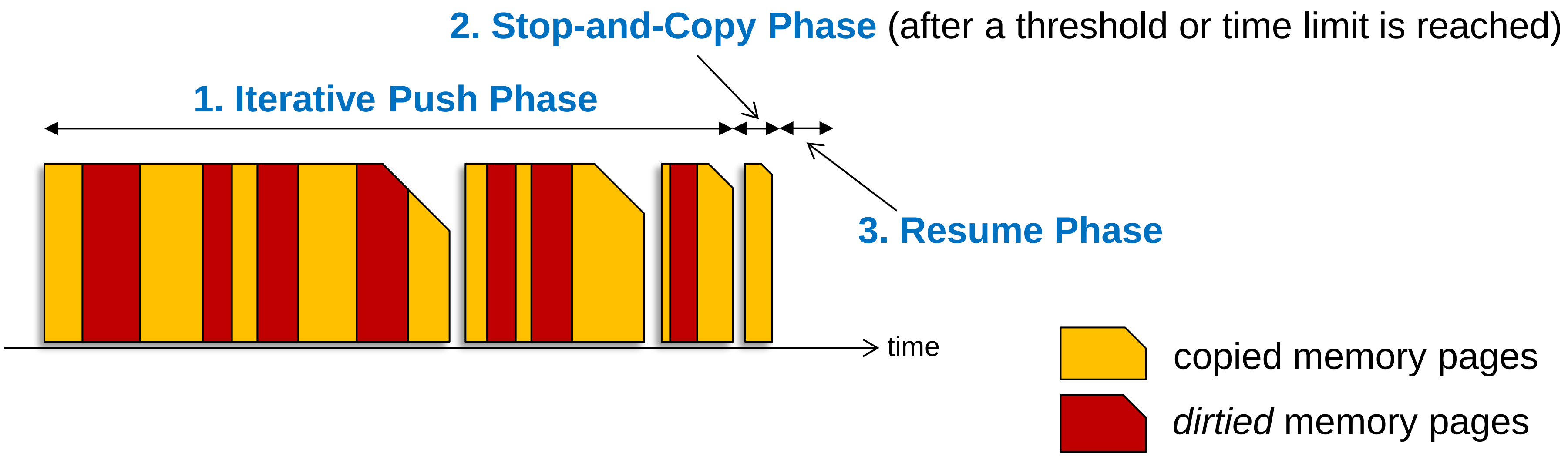
Live Migration of Virtual Machines

- Moving services from one host/DC to another with **minimal disruption** to end-user **service availability**
- Current state of VM's kernel and running processes must be maintained
  - **storage state migration** through NAS synchronization
    - bulk data transfers to copy disk image (before migration starts)
    - **copy-on-write** mechanisms applied to template disk images allows to copy only the differences (live block migration)
  - **network state migration** to maintain connections
    - **IP identifier/locator split principle** solutions: HIP, ILNP, LISP
    - **Software Defined Networking** technologies to dynamically reroute traffic by programming the forwarding paths

- Focus on **memory state migration**

Live Migration of Virtual Machines

- Two approaches for memory state migration
  - pre-copy: push most of the memory pages to destination host before stopping VM at source host
  - post-copy: pull most of the memory pages from source host after resuming VM at destination host
- We assume the **pre-copy** approach
  - adoped by Xen, KVM, VirtualBox, etc.



2. Stop-and-Copy Phase (after a threshold or time limit is reached)

1. Iterative Push Phase

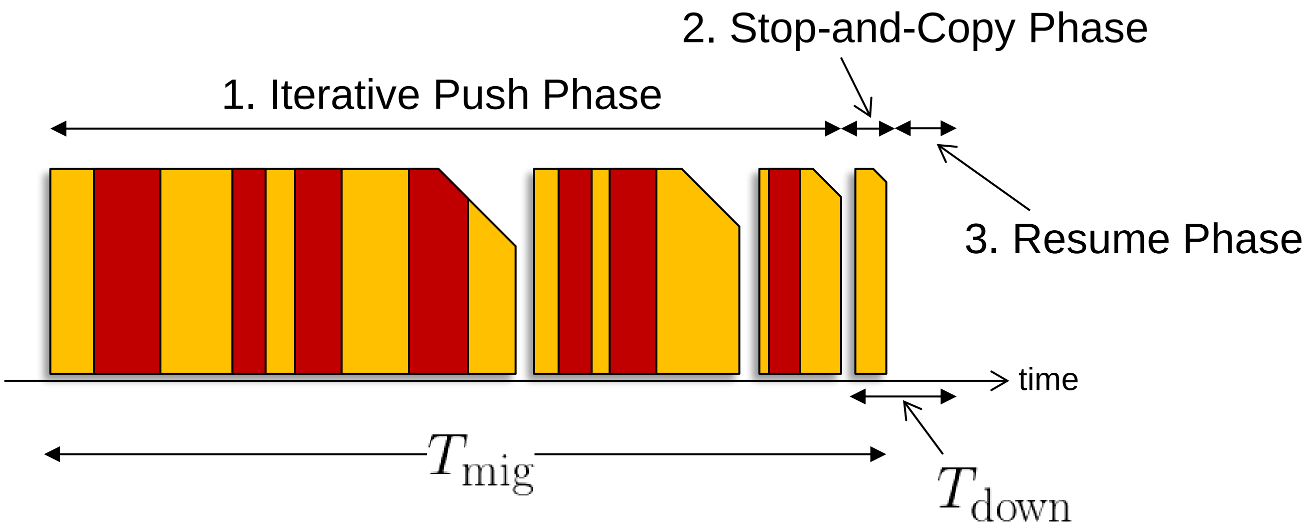
3. Resume Phase

🟨 copied memory pages

🟥 *dirtied* memory pages

Performance Metrics for VM Live Migration

- **Downtime** (T_{down}): amount of time the VM is suspended
  - measures the end-user's perceived quality
- **Total Migration Time** (T_{mig}): amount of time needed to copy the whole memory
  - measures the impact of the migration process on both communication infrastructure and DC capacity
  - network and computing resources **busy during whole migration time**



Simplified Model of VM Live Migration [8]

- $V_0(z)$ size of memory allotted to VM z to be migrated
- $D(z) = D$ all VMs show the same fixed page dirtying rate
- $P(z) = P$ all VMs have the same memory page size
- $b(z) = b$ the bit rate used to migrate each VM is guaranteed
- $\gamma = (P\,D)/b < 1$ condition for pre-copy algorithm to be sustainable

$$T_{\text{mig}}(z) = \sum_{i=0}^{I(z)} T_i(z) = \frac{V_0(z)}{b}\,\frac{1-\gamma^{I(z)+1}}{1-\gamma}$$

$$T_{\text{down}}(z) = \frac{V_0(z)}{b}\,\gamma^{I(z)}$$

dirty memory size threshold

number of iterations

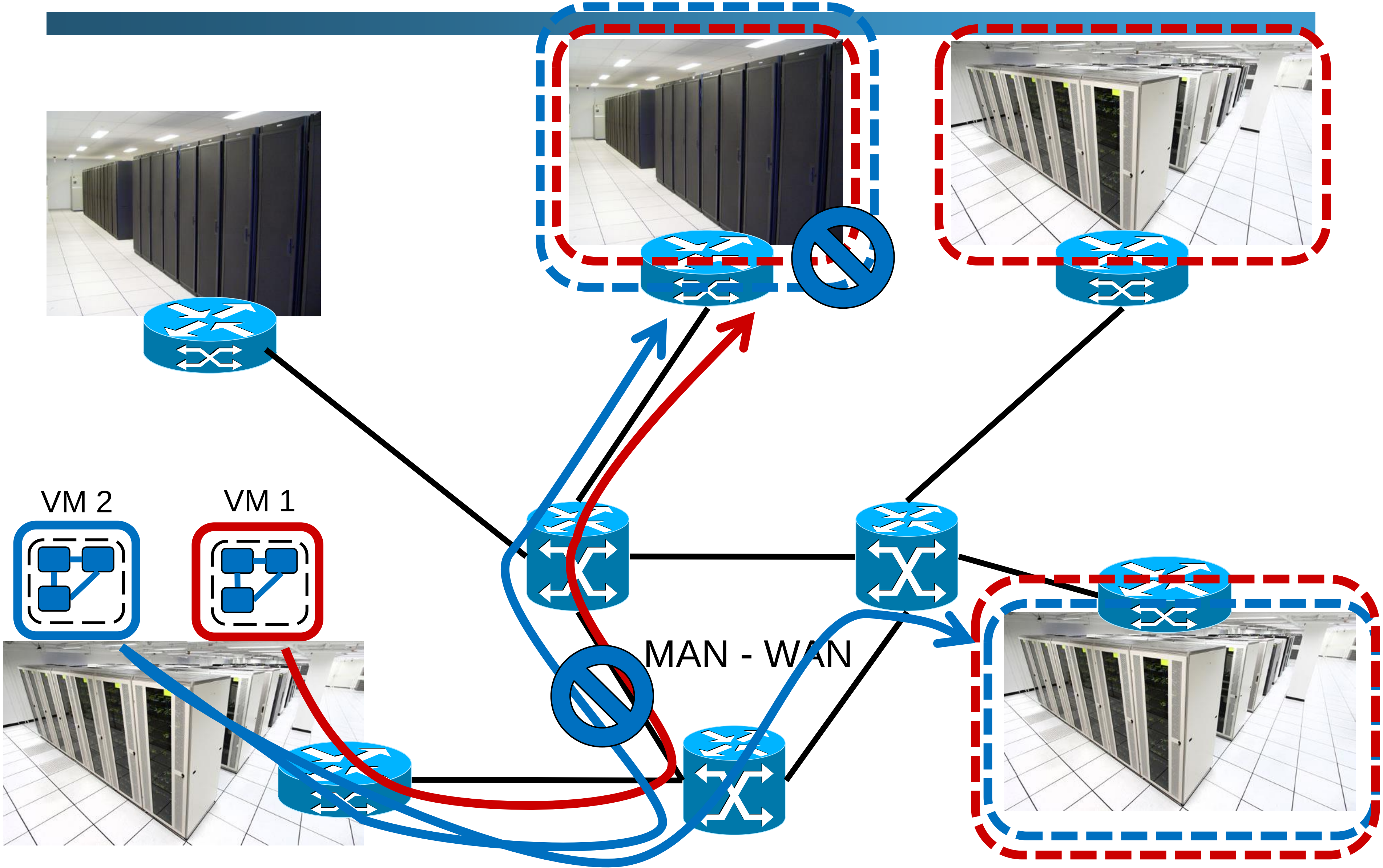
$$I(z) = \min\left\{\left\lceil \log_{\gamma}\frac{V_{\text{th}}}{V_0(z)}\right\rceil, I_{\max}\right\}$$

max no. of iterations

Federated Cloud Network Scenario

- Federated DCs are interconnected by a full mesh of guaranteed-bandwidth network pipes
  - pre-established MPLS LSPs between edge routers
  - pre-established lightpaths on optical inter-DC network
- Workload of VM migrating from source DC can be hosted by a subset of remote federated DCs
  - suitable hypervisor/storage resource available in some DCs only
  - service-specific DC location constraints (e.g., due to latency)
  - other constraints due to load balancing, energy savings, etc.
- Available remote DC resources assigned following the **anycast** service model
  - any DC in the available/suitable subset is equivalent for hosting the VM to be migrated

Federated Cloud Network Scenario



Federated Cloud Network Model Assumptions

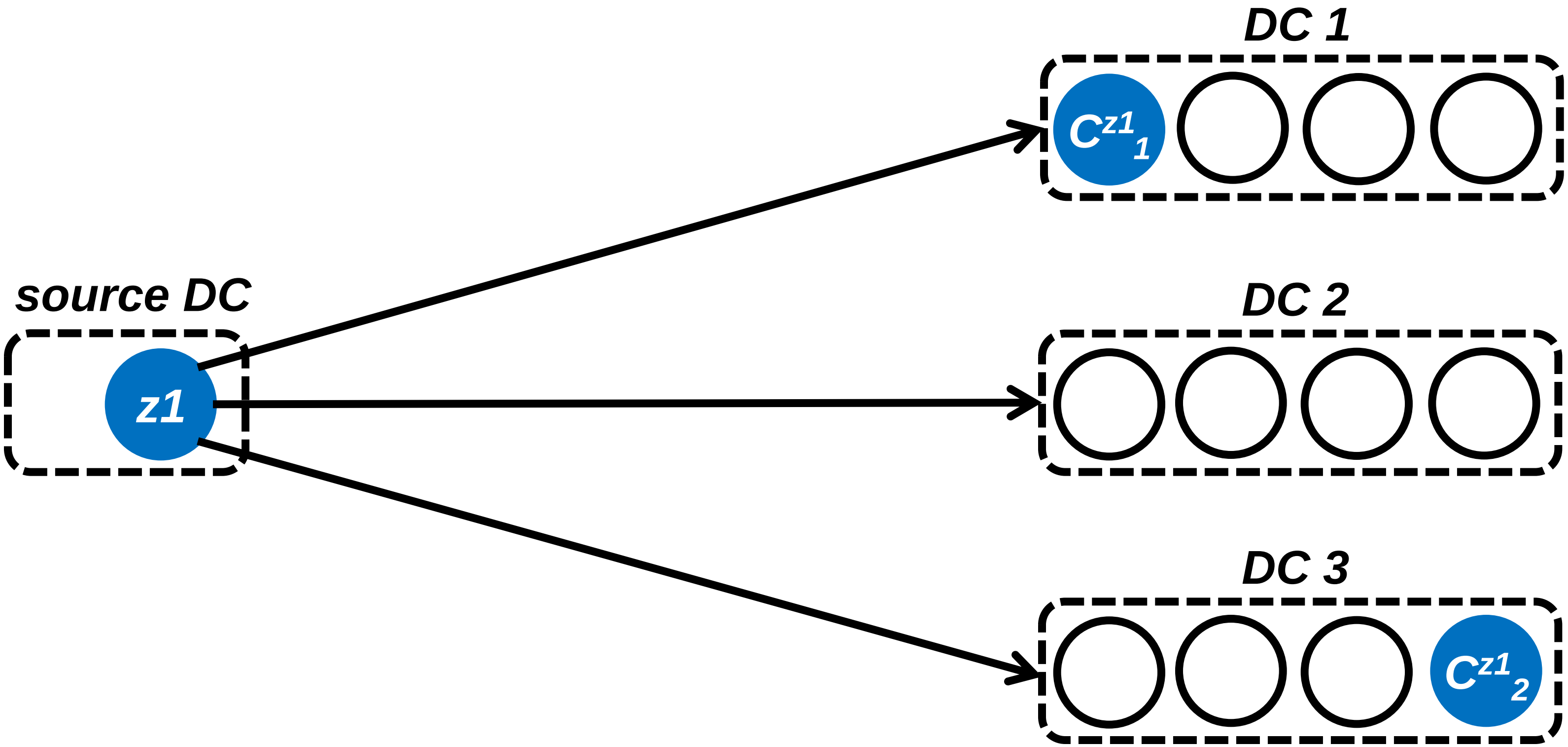
- **A.1:** each VM migration consumes the same amount of channel capacity ***b***
- **A.2:** each network pipe provides the same total amount of guaranteed capacity ***B***
- **A.3:** each remote DC has the computing and storage capacity of hosting up to ***k*** VMs
- **A.4:** each migration request is allowed to choose among ***m*** instances of the requested computing/storage resources, which are randomly distributed over the ***n*** remote DCs
  - considering the general case when multiple instances of the same resources can be available in the same DC
- **A.5:** resource state, as seen by a given DC, is related to the number of ongoing/completed VM migrations originated by that DC
  - network state = no. of busy pipes: $r = 0, 1, \ldots, n\lfloor B/b \rfloor$
  - DC state = no. of busy computing resources: $r' = 0, 1, \ldots, nk$

Federated Cloud Network Model

Example with $n = 3$, $k = 4$, $m = 2$, $b = B$

Network state: $r = 0$

DC state: $r' = 0$

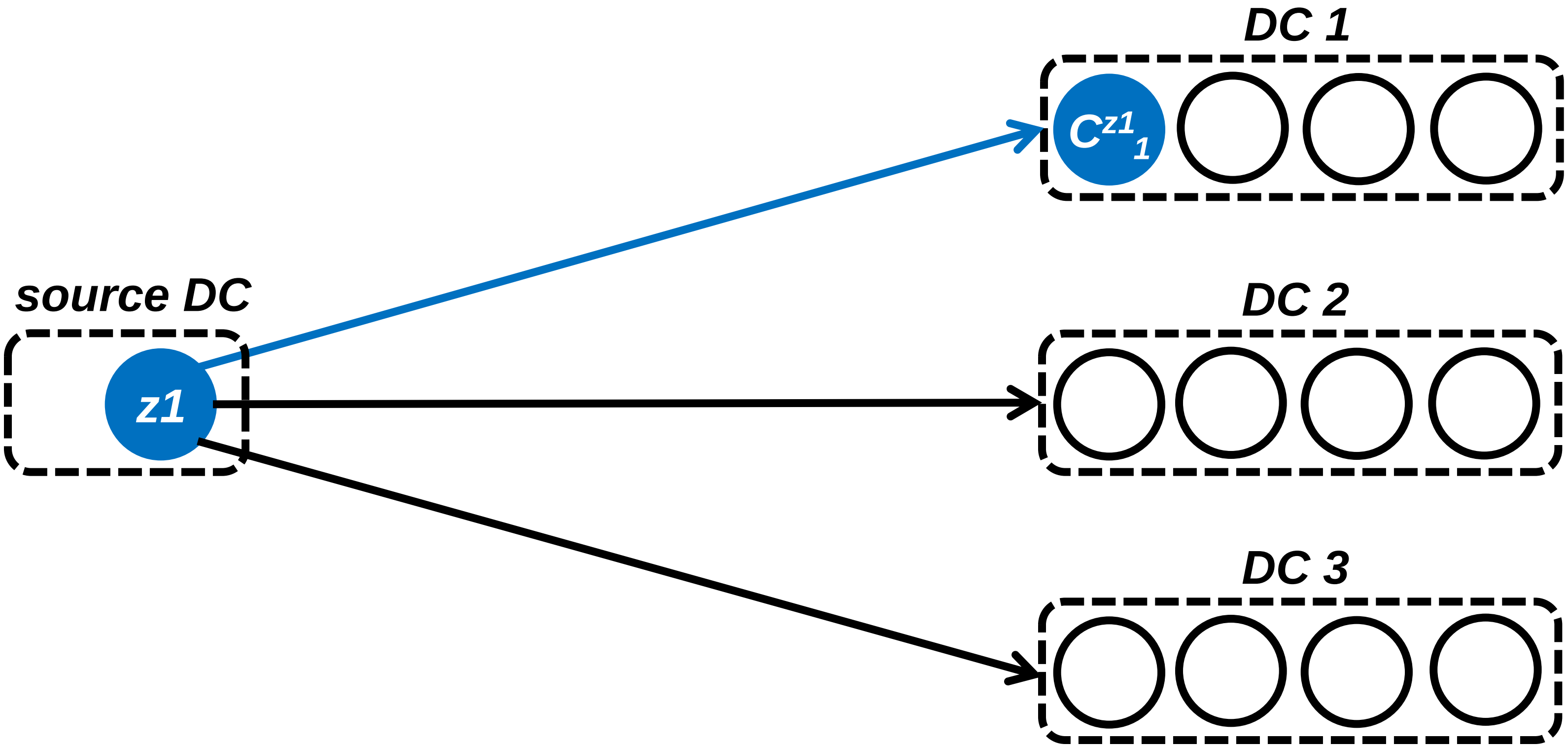


Federated Cloud Network Model

Example with $n = 3$, $k = 4$, $m = 2$, $b = B$

Network state: $r = 1$

DC state: $r' = 1$

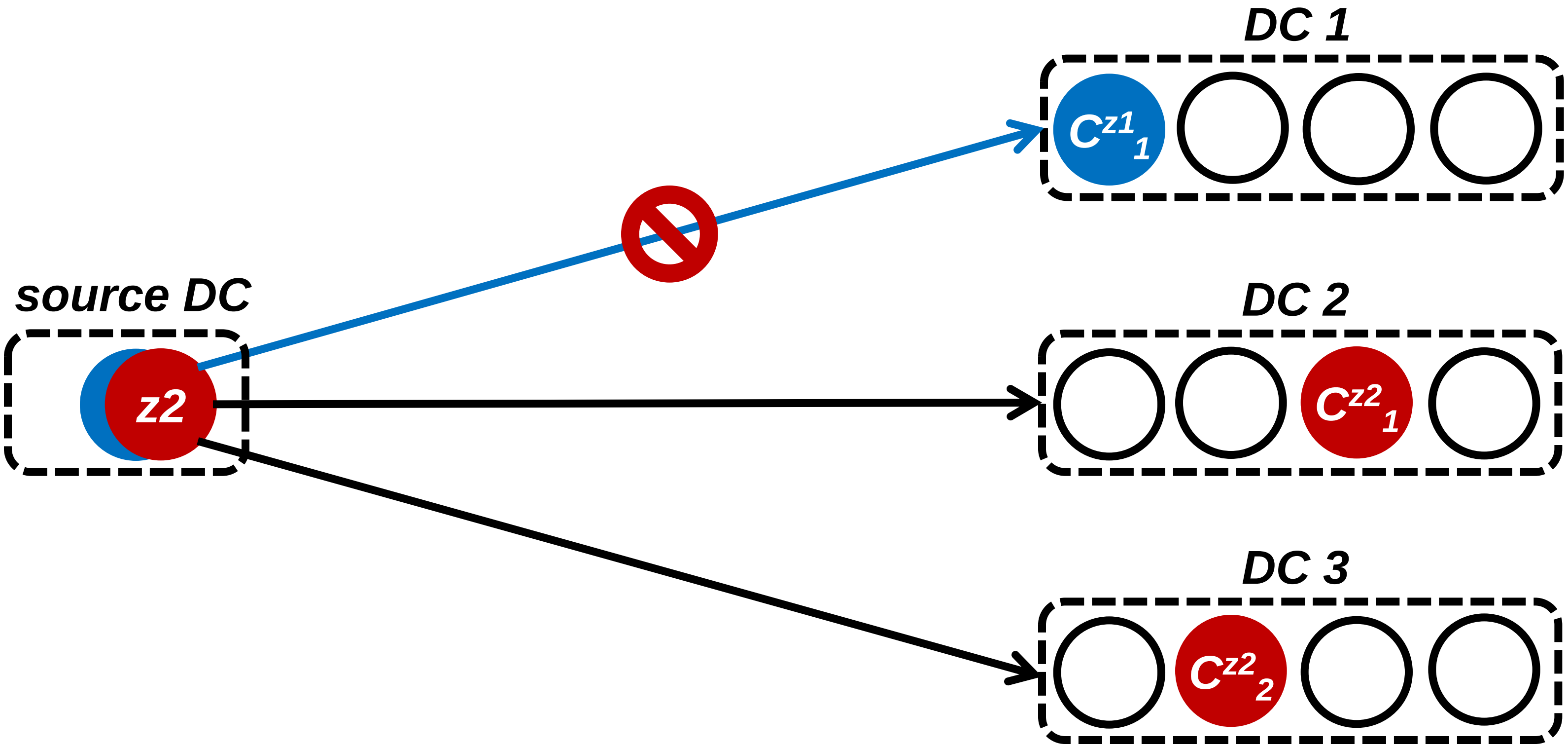


Federated Cloud Network Model

Example with n = 3 , k = 4, m = 2, b = B

Network state: r = 1

DC state: r' = 1

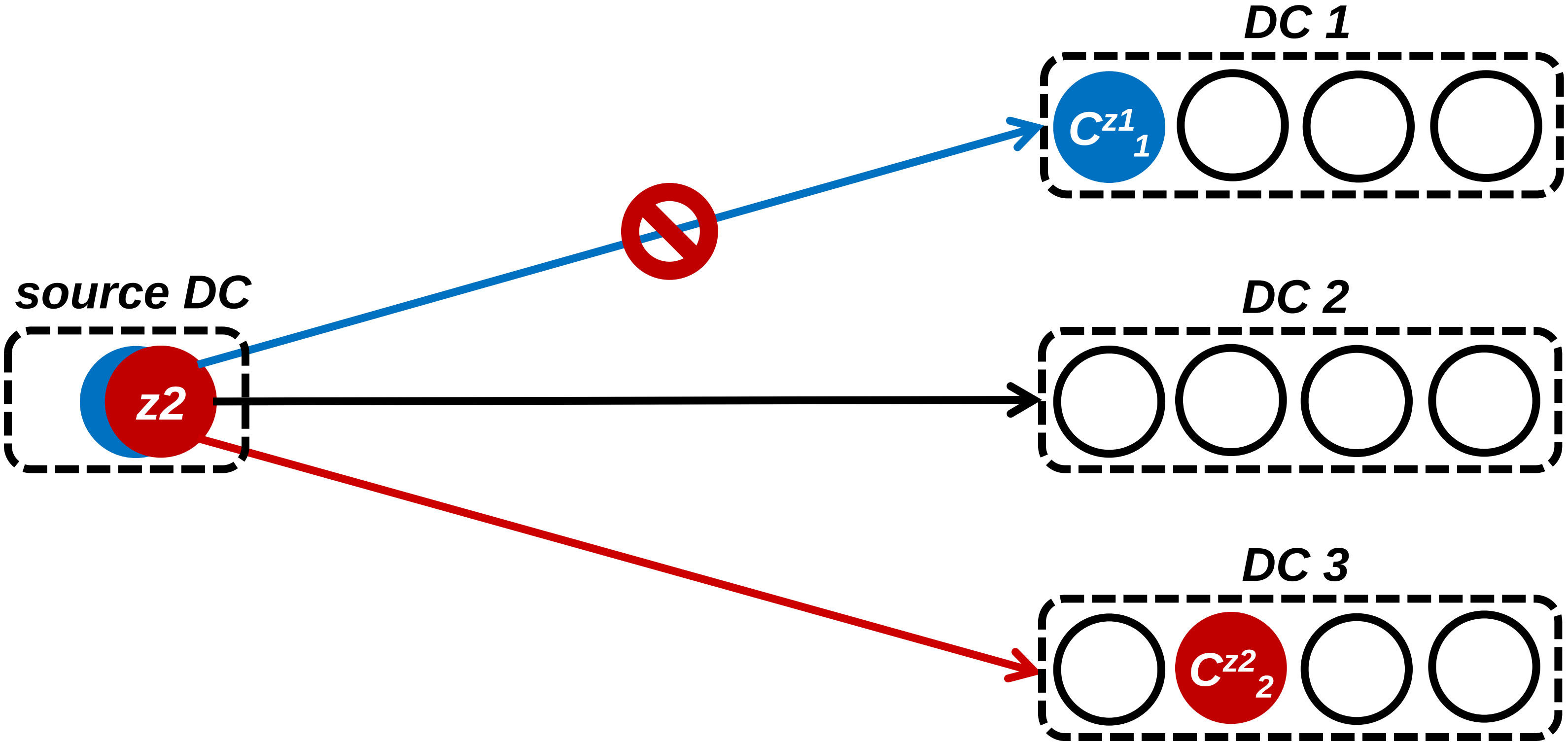


Federated Cloud Network Model

Example with $n = 3$, $k = 4$, $m = 2$, $b = B$

Network state: $r = 2$

DC state: $r' = 2$

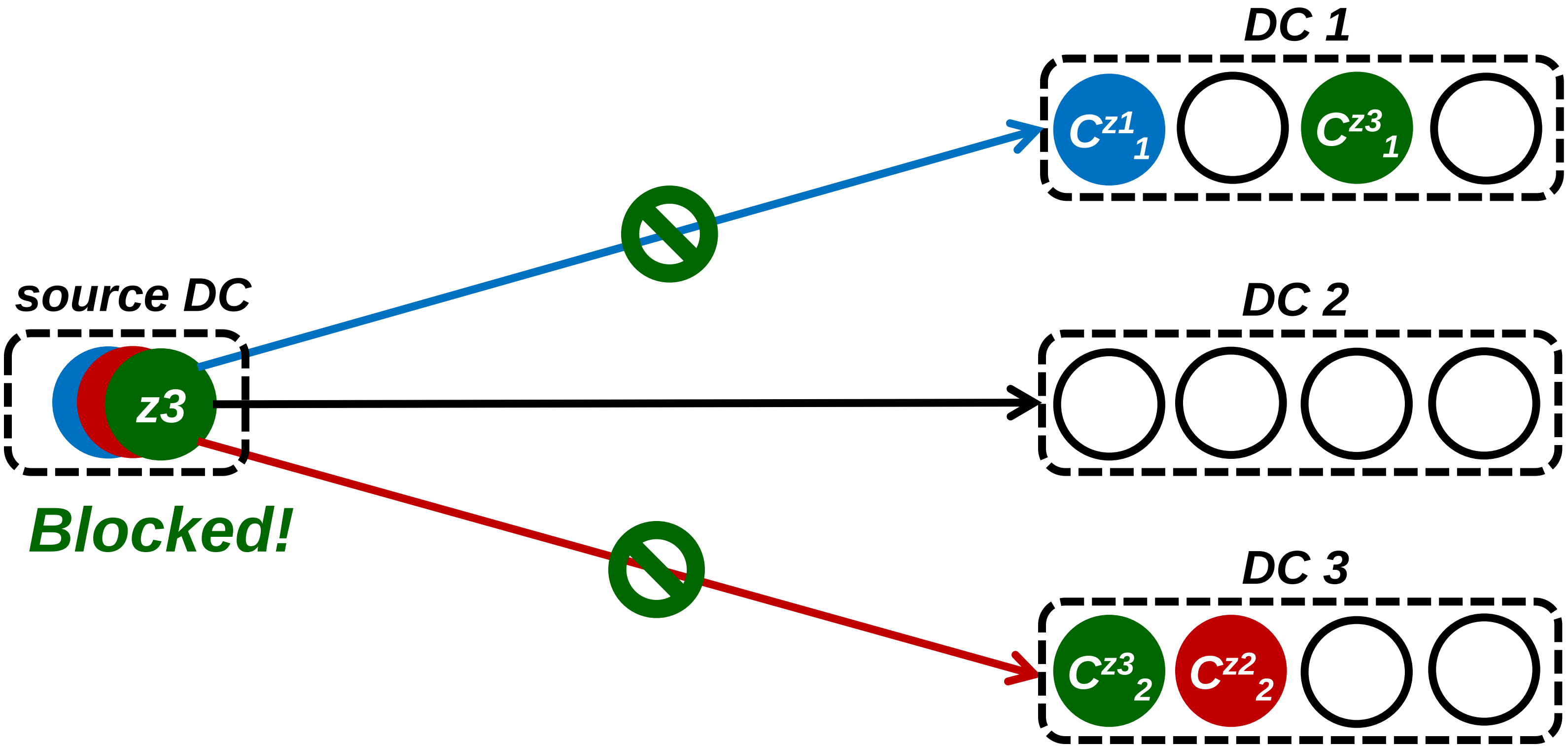


Federated Cloud Network Model

Example with $n = 3$, $k = 4$, $m = 2$, $b = B$

Network state: $r = 2$

DC state: $r' = 2$

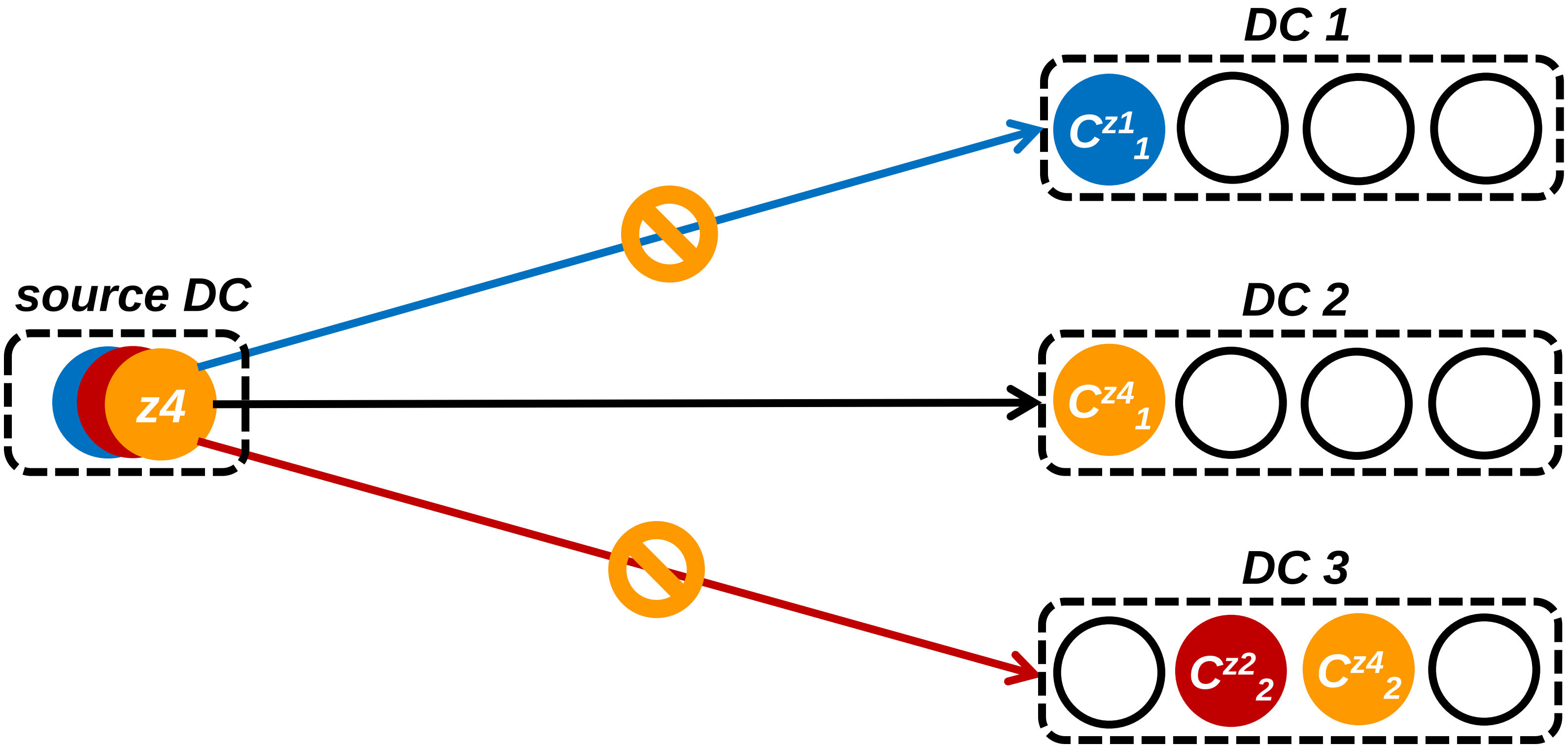


Federated Cloud Network Model

Example with $n = 3$, $k = 4$, $m = 2$, $b = B$

Network state: $r = 2$

DC state: $r' = 2$

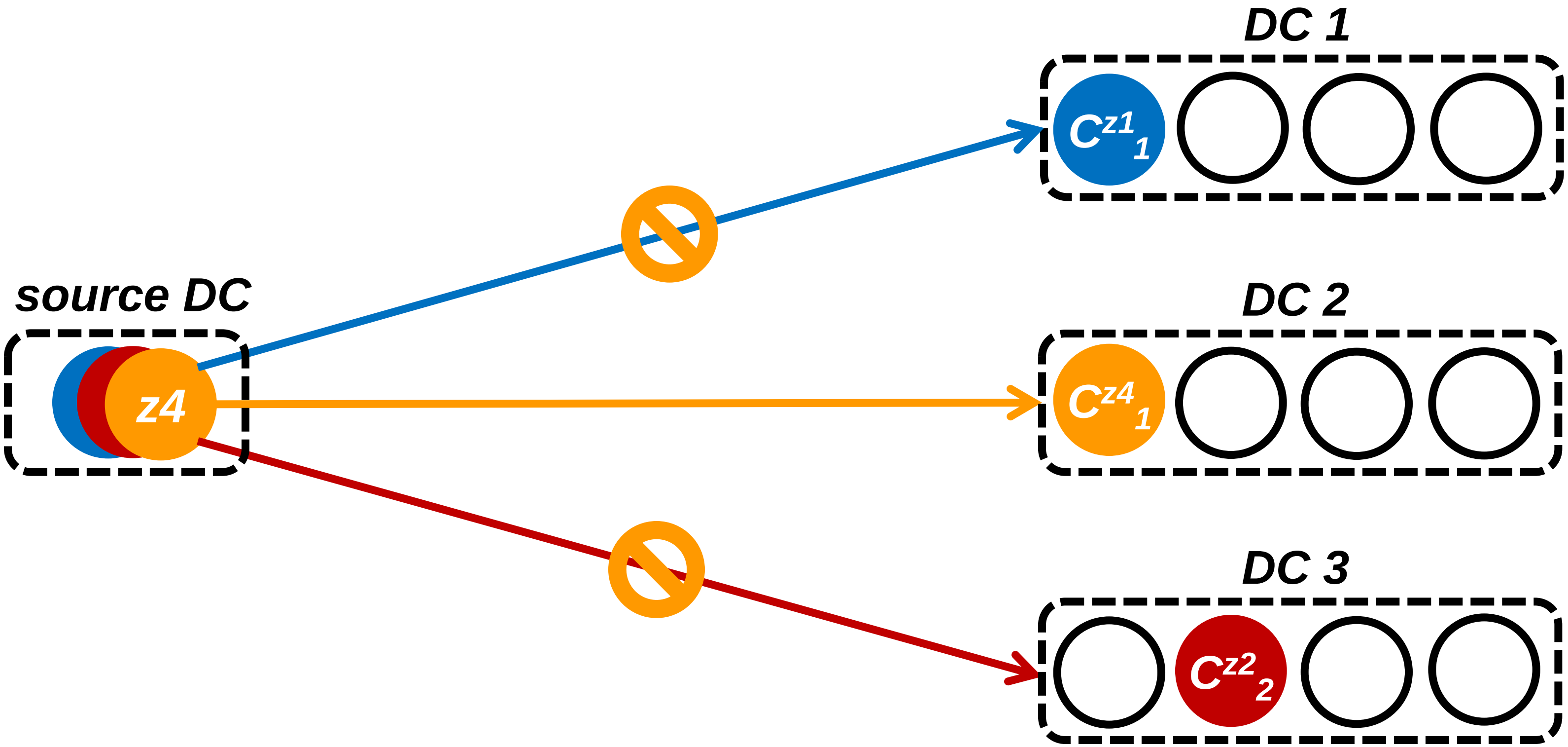


Federated Cloud Network Model

Example with $n = 3$, $k = 4$, $m = 2$, $b = B$

Network state: $r = 3$

DC state: $r' = 3$

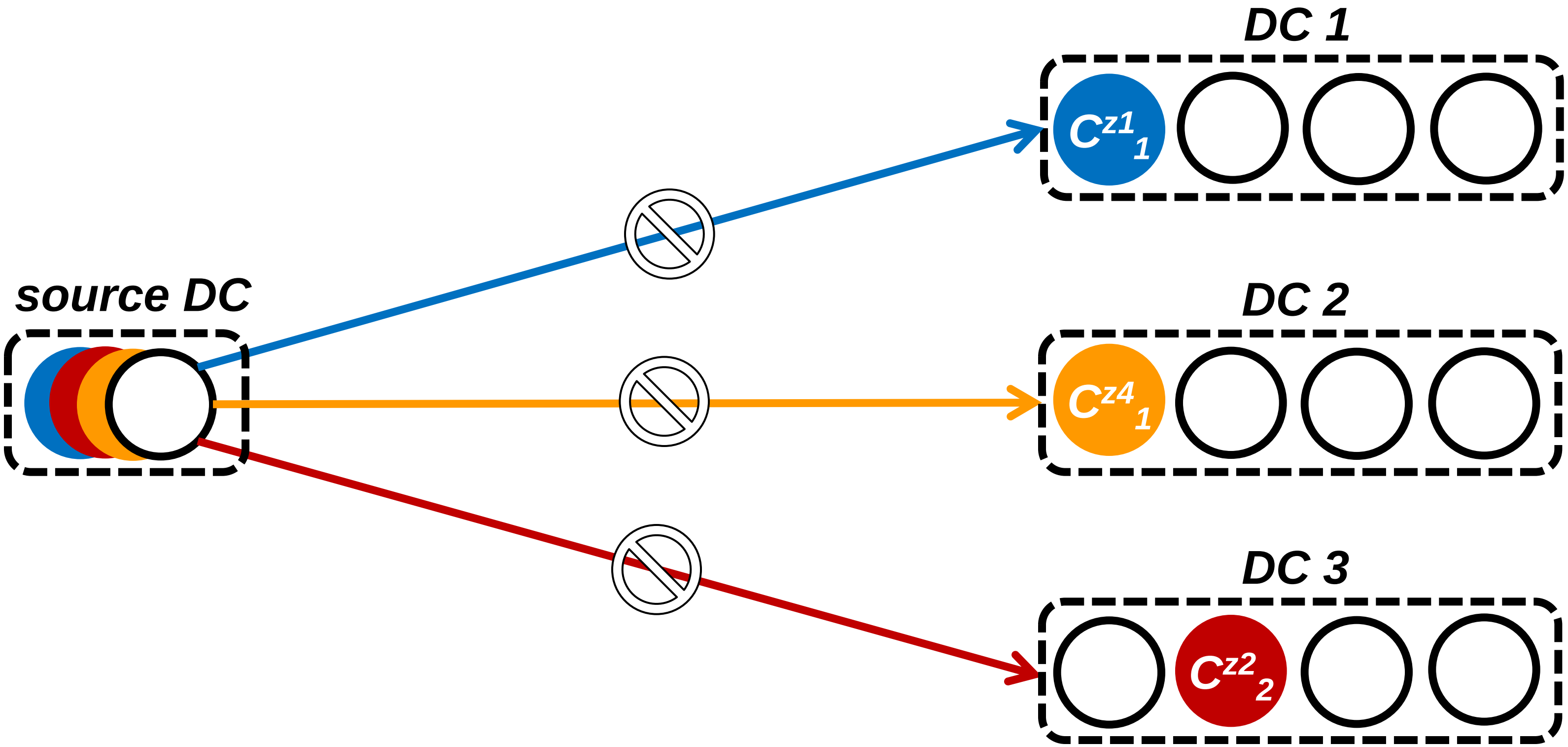


Federated Cloud Network Model

Example with $n = 3$, $k = 4$, $m = 2$, $b = B$

Network state: $r = 3$

DC state: $r' = 3$

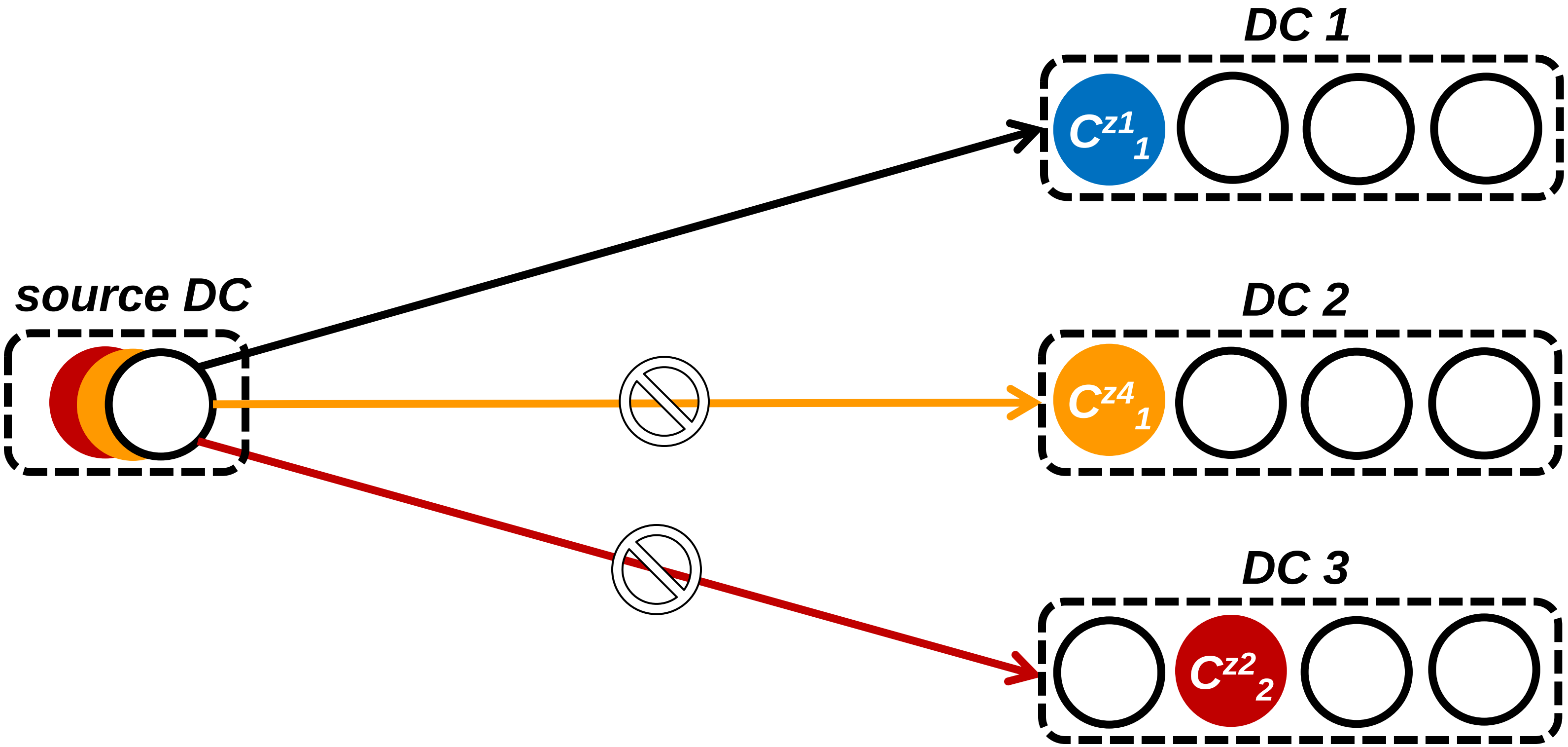


Federated Cloud Network Model

Example with $n = 3$, $k = 4$, $m = 2$, $b = B$

Network state: $r = 2$

DC state: $r' = 3$

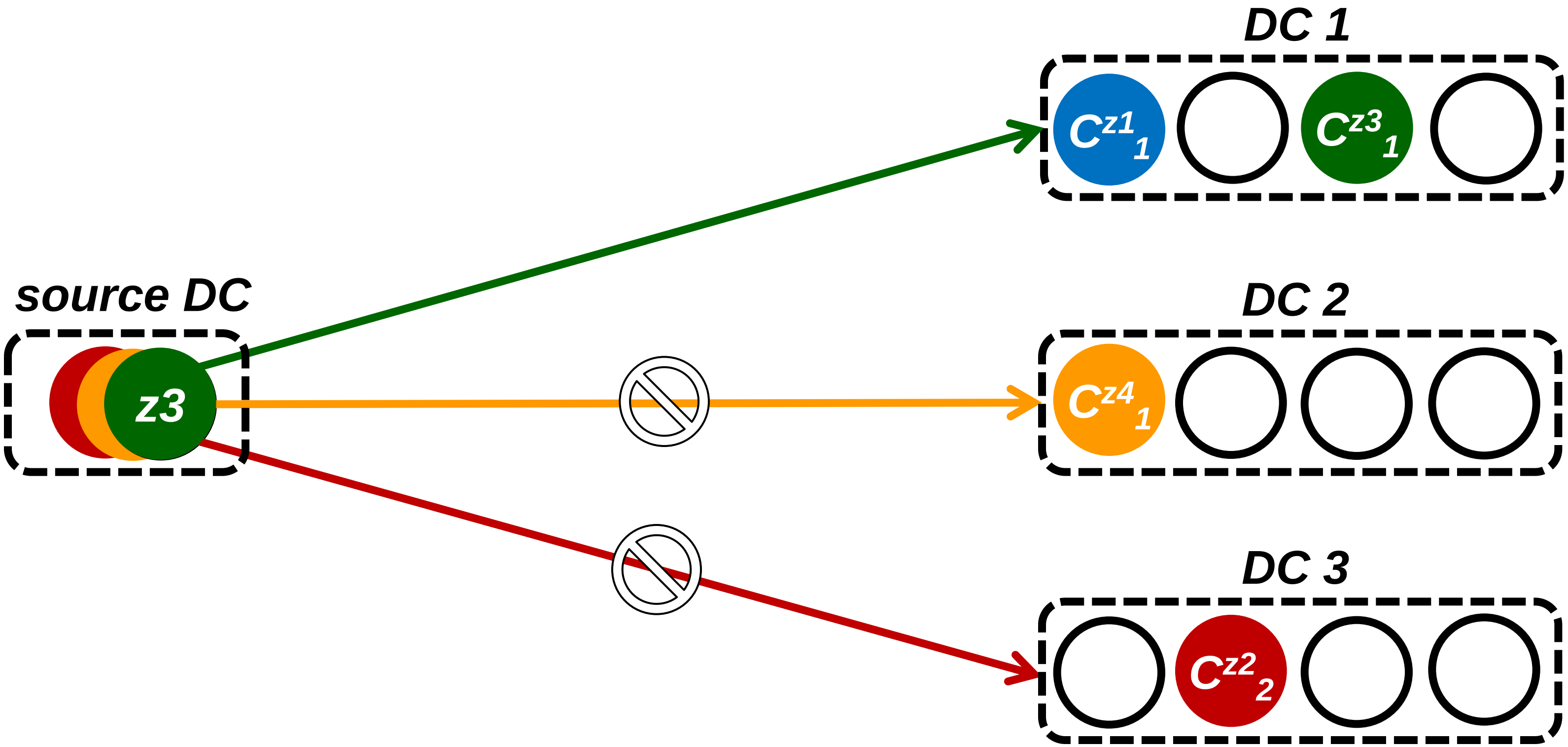


Federated Cloud Network Model

Example with *n* = 3 , *k* = 4, *m* = 2, *b* = *B*

Network state: *r* = 3

DC state: *r’* = 4

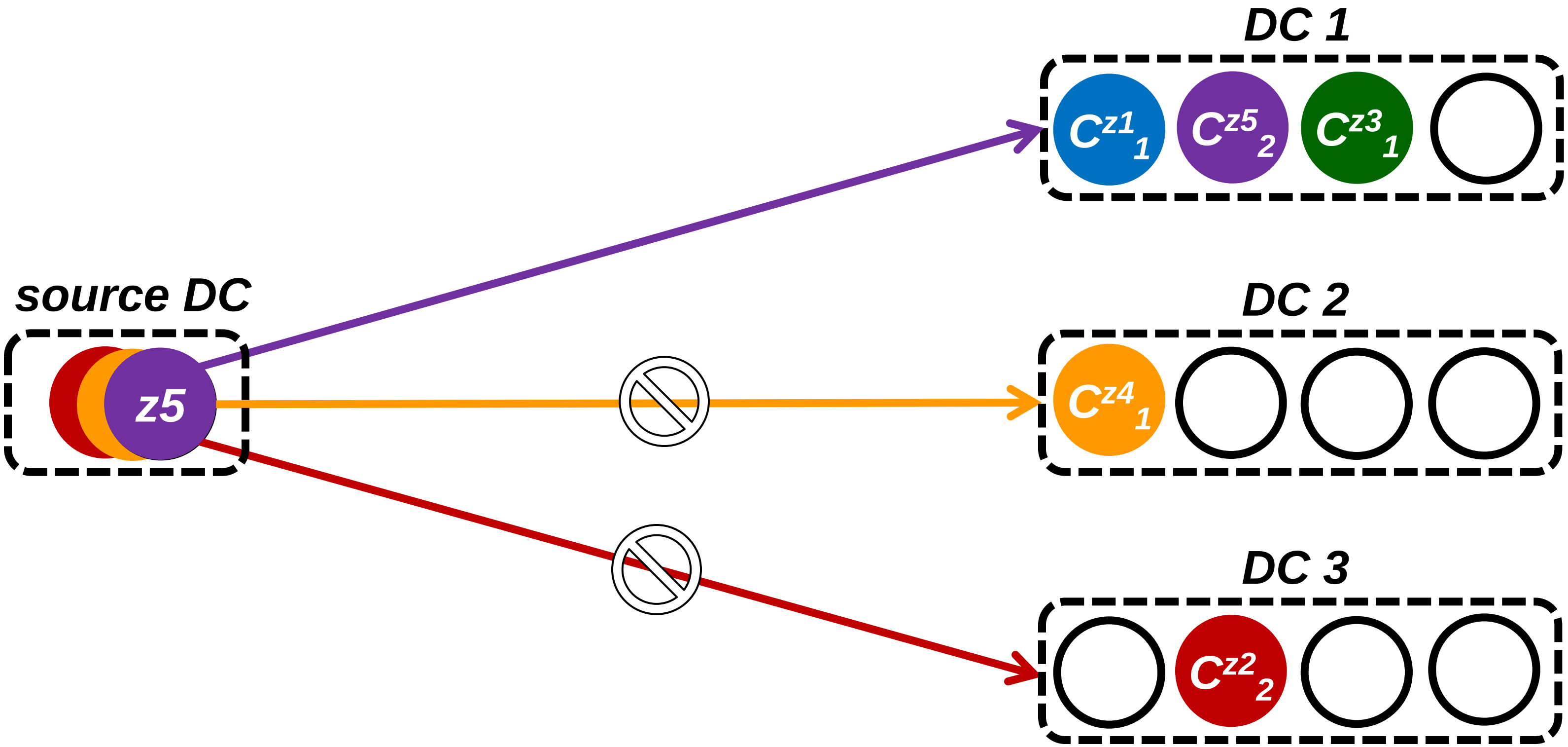


Federated Cloud Network Model

Example with $n = 3$, $k = 4$, $m = 2$, $b = B$

Network state: $r = 3$

DC state: $r' = 5$

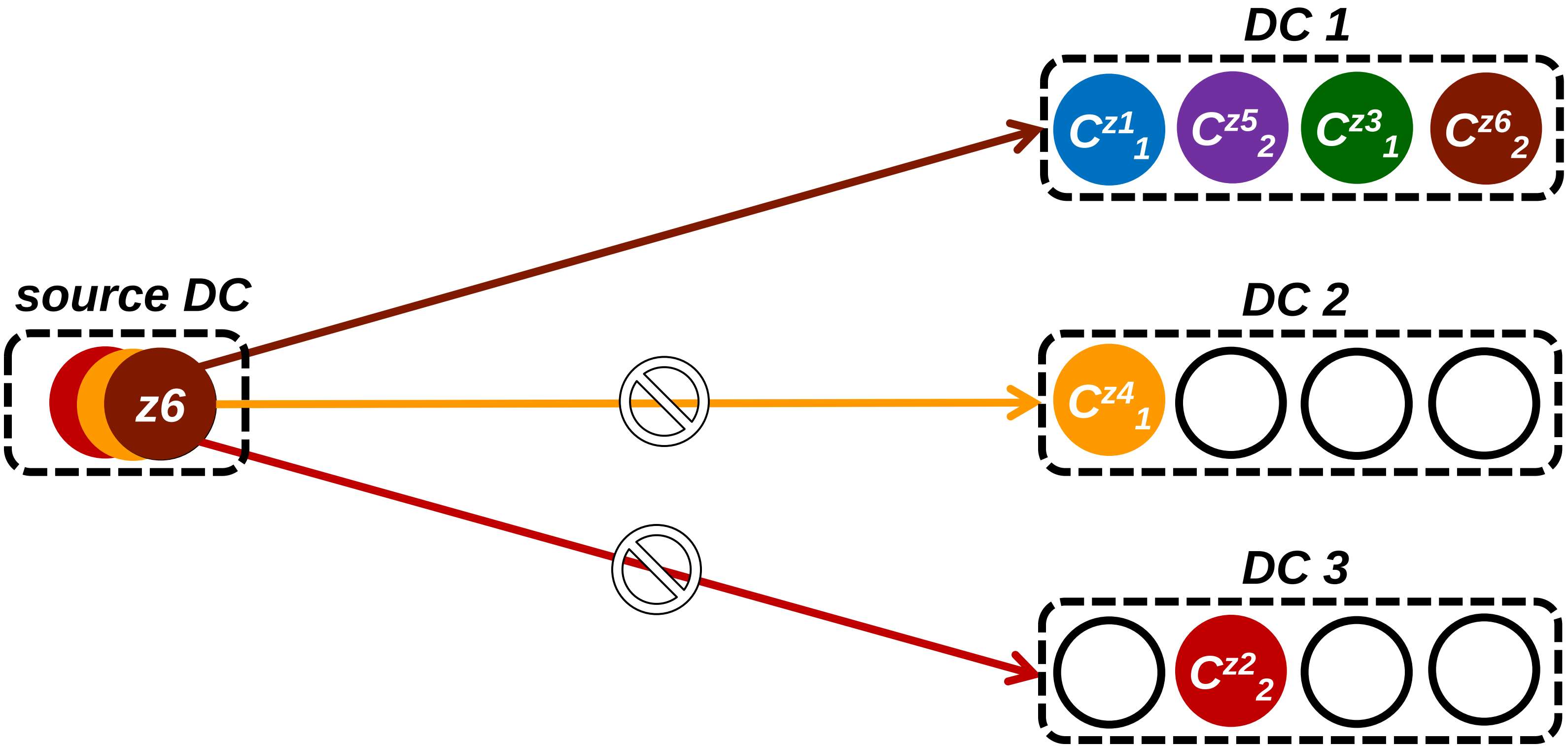


Federated Cloud Network Model

Example with $n = 3$, $k = 4$, $m = 2$, $b = B$

Network state: $r = 3$

DC state: $r' = 6$

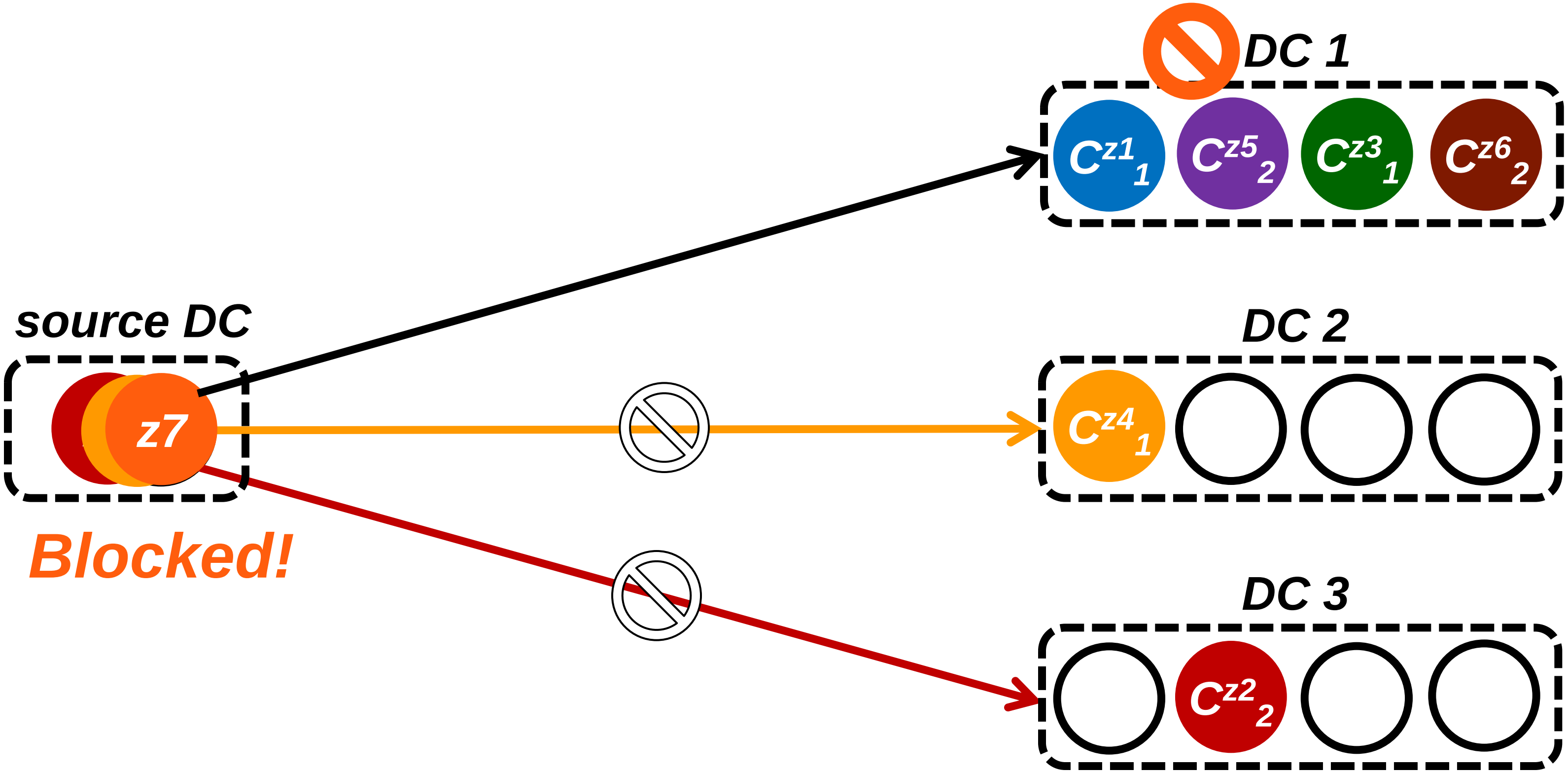


Federated Cloud Network Model

Example with $n = 3$, $k = 4$, $m = 2$, $b = B$

Network state: $r = 2$

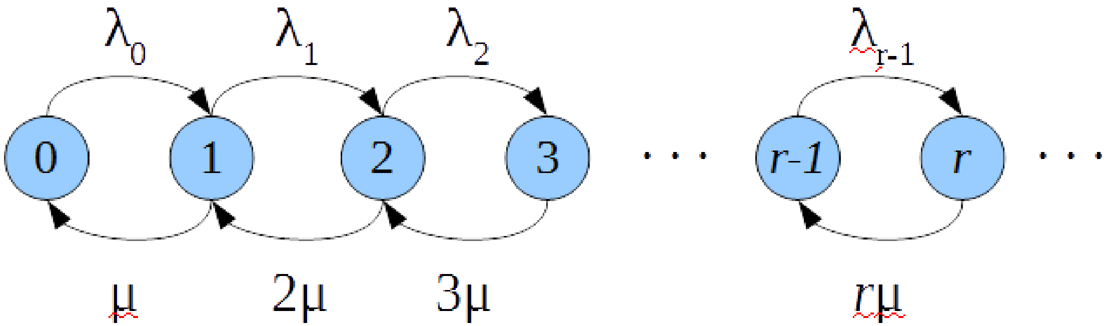
DC state: $r' = 6$



Markovian Model of Resource Allocation

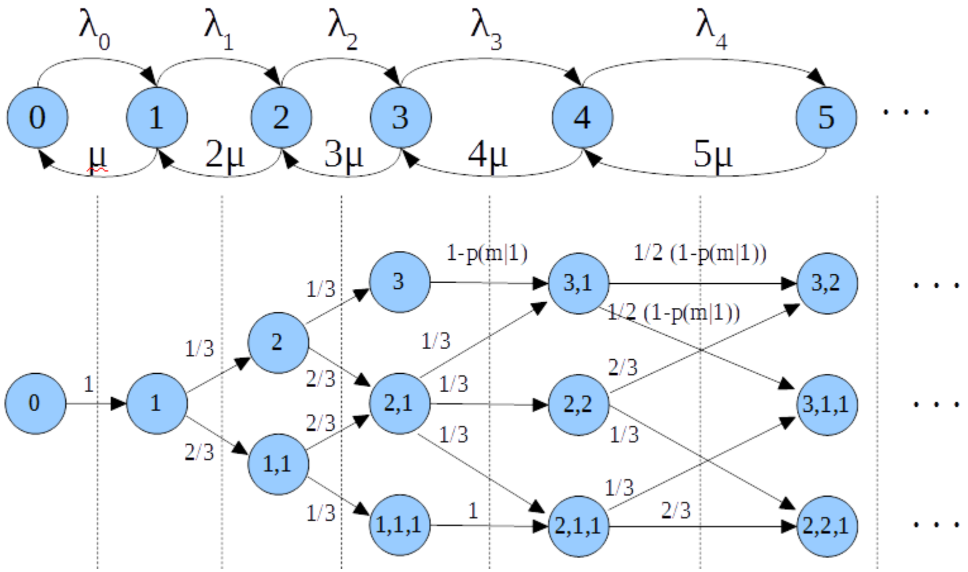
- VM migration requests as a Poisson process
  - request arrival rate λ

- Service rate is the reciprocal of the average resource renewal time
  - network: $\mu_{\mathrm{NET}} = 1/E\left[T_{\mathrm{mig}}(z)\right]$
  - DC: μ_{DC}
  - offered load: $A_0 = \lambda/\mu$
  - **loss system**: results valid for any service time distribution with finite mean



Approximate Sub-state Probabilities

- Given state *r*, many combinations of resource allocation are possible
- Exact solution would require to compute all sub-states probabilities
- Approximate solution with reduced state space considering only "forward" state evolution
- Recursive expression of sub-space probabilities $s(c_1, c_2, \ldots, c_n \,|\, r)$



$$s(3,1|4) = \frac{(1-p(m|1))\, s(3|3) + 1/3\, s(2,1|3)}{(1-p(m|1))\, s(3|3) + s(2,1|3) + s(1,1,1|3)}$$

n = 3, *B* = 3*b*

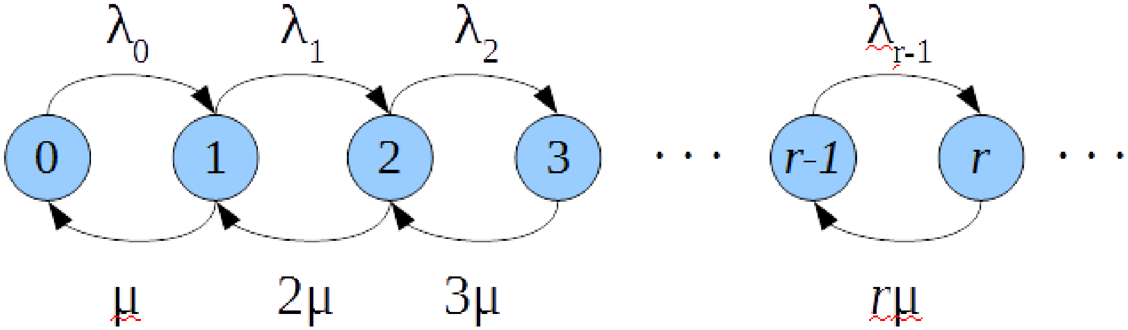
Prob. that *m* suitable resources are hosted by unreachable or busy DCs:

$$p(m|\ell) = \prod_{i=0}^{m-1} \frac{\ell k - i}{nk - i} \qquad \ell = 1, \ldots, n$$

Prob. request blocked in state 5:

$$P_{B|5} = p(m|1)\, s(3,2|5) + p(m|1)\, s(3,1,1|5)$$

Steady-State Probabilities



$$
\begin{aligned}
\lambda_0 &= \lambda \\
\lambda_r &= \left(1 - P_{\mathrm{B}|r}\right)\lambda\,, \quad r = 1, 2, \ldots, r_{\max} - 1
\end{aligned}
$$

$$
\begin{aligned}
P_0 &= \left(1 + A_0 + \sum_{r=2}^{r_{\max}} \prod_{\ell=1}^{r-1} \left(1 - P_{\mathrm{B}|\ell}\right) \frac{A_0^r}{r!}\right)^{-1} \\
P_1 &= P_0 A_0 \\
P_r &= P_0 \prod_{\ell=1}^{r-1} \left(1 - P_{\mathrm{B}|\ell}\right) \frac{A_0^r}{r!} \quad 2 \le r \le r_{\max}
\end{aligned}
$$

Blocking probability:

$$
P_{\mathrm{B}} = \sum_{r=1}^{r_{\max}} P_{\mathrm{B}|r} P_r
$$

Combining the Two Resource States

- Any migration request blocked due to lack of computing resources will not consume network resources
- Actual load on network resources: $A_{0,\mathrm{NET}} = \frac{\lambda}{\mu_{\mathrm{NET}}}(1 - P_{\mathrm{B,DC}})$
- Total blocking probability:

$$P_{\mathrm{B,TOT}} = P_{\mathrm{B,DC}} + P_{\mathrm{B,NET}} - P_{\mathrm{B,DC}}\, P_{\mathrm{B,NET}}$$

Numerical Results

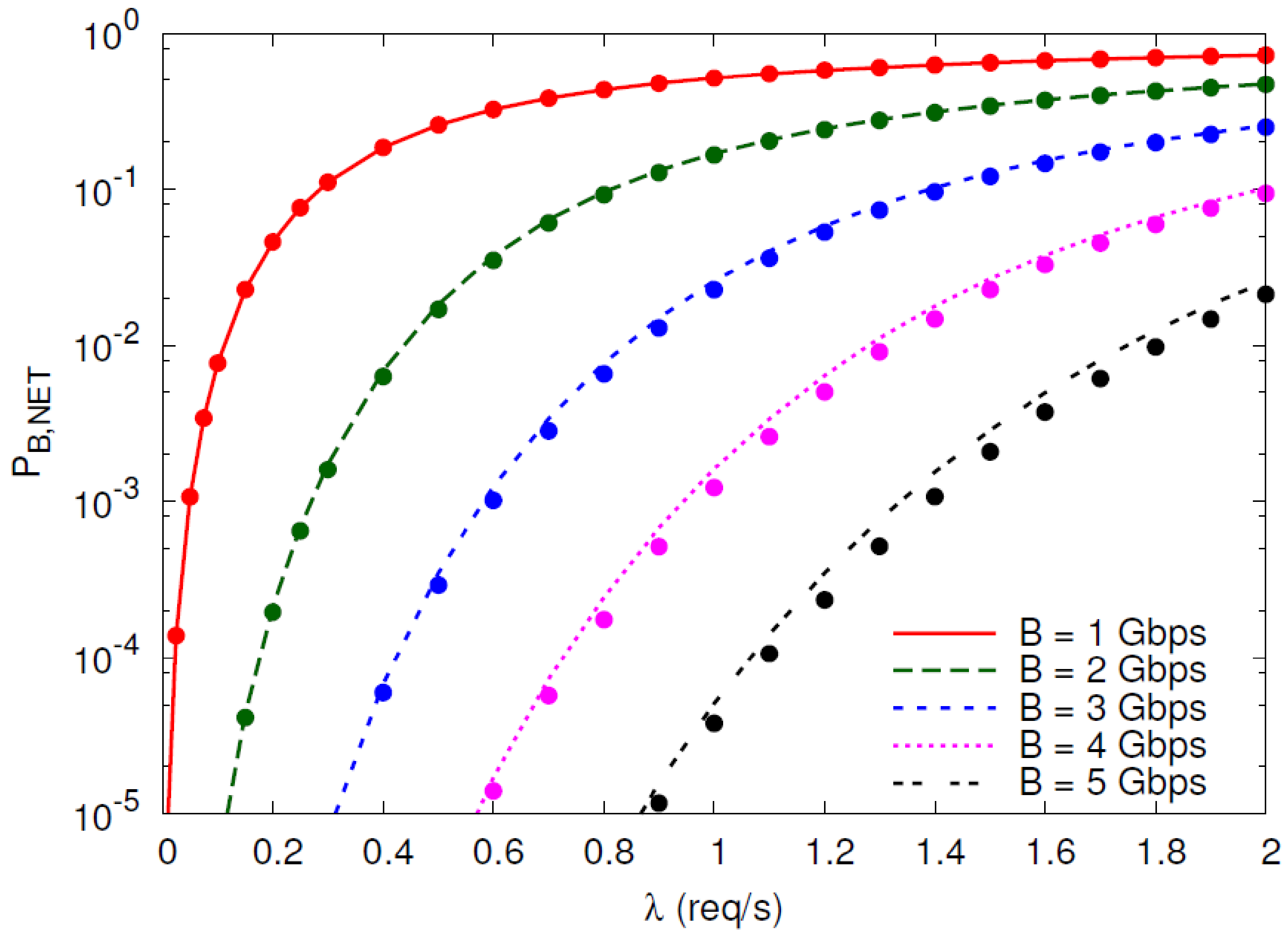
- VM memory size distribution
  - bimodal distribution: large and small VMs
  - $V_0(z) = V_0$ with probability 75%
  - $V_0(z) = 4\,V_0$ with probability 25%

- Reference values for model parameters

$b = 1\,\mathrm{Gbps}$	$P = 4\,\mathrm{KB}$	$B = 3\,\mathrm{Gbps}$
$\mu_{\mathrm{DC}} = \mu_{\mathrm{NET}}/5$	$V_{\mathrm{th}} = 100\,\mathrm{MB}$	$n = 5$
$V_0 = 512\,\mathrm{MB}$	$I_{\mathrm{max}} = 8$	$m = 3$
$D = 2500\,\mathrm{pps}$	$T_{\mathrm{res}} = 100\,\mathrm{ms}$	$k = 8$

- Model curves + simulation points to validate model accuracy

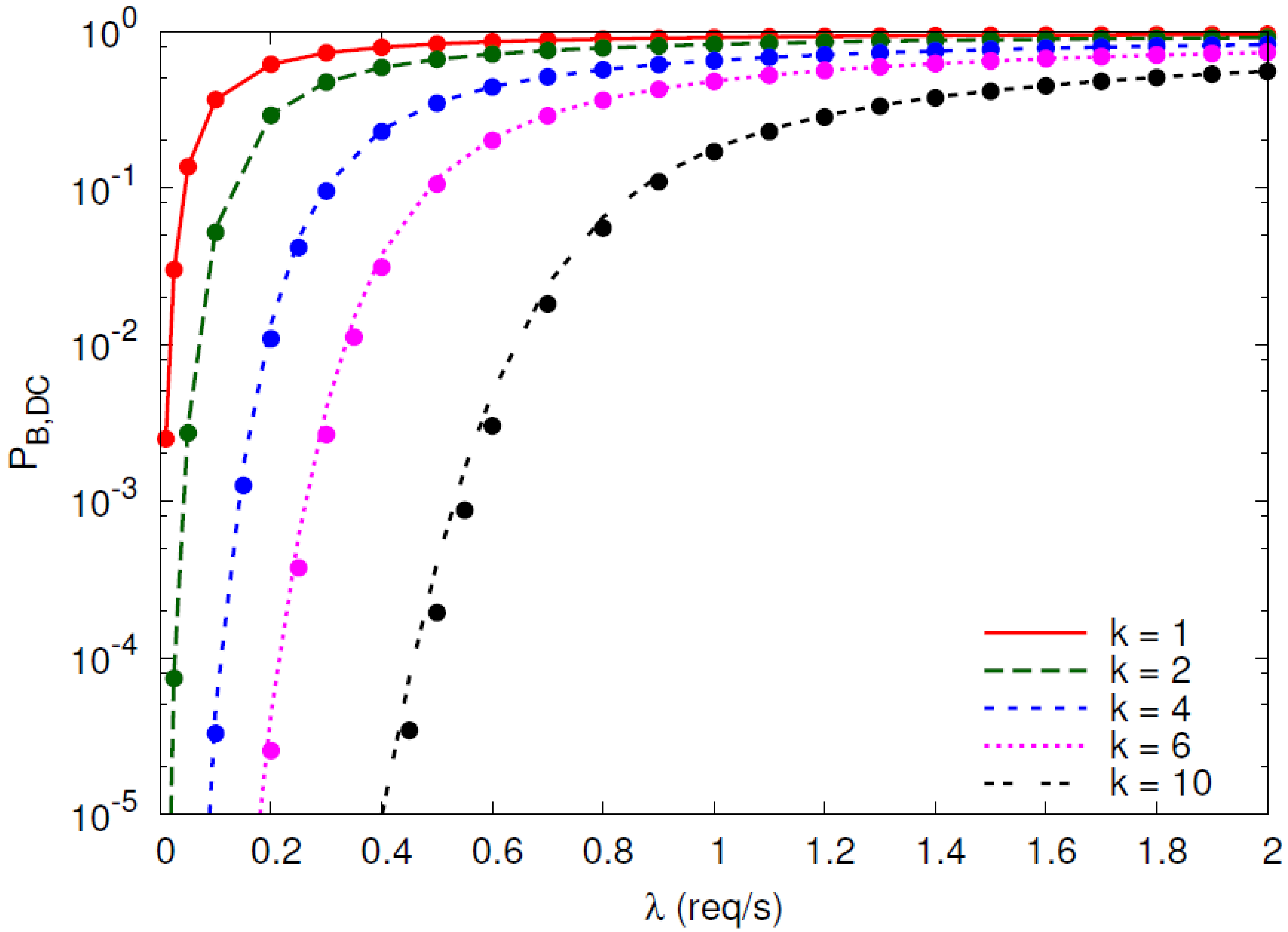
Impact of Network Resource Sharing



$k = \infty$

- Good match with simulations → reasonable accuracy
- Model allows to dimension the cloud federation network capacity

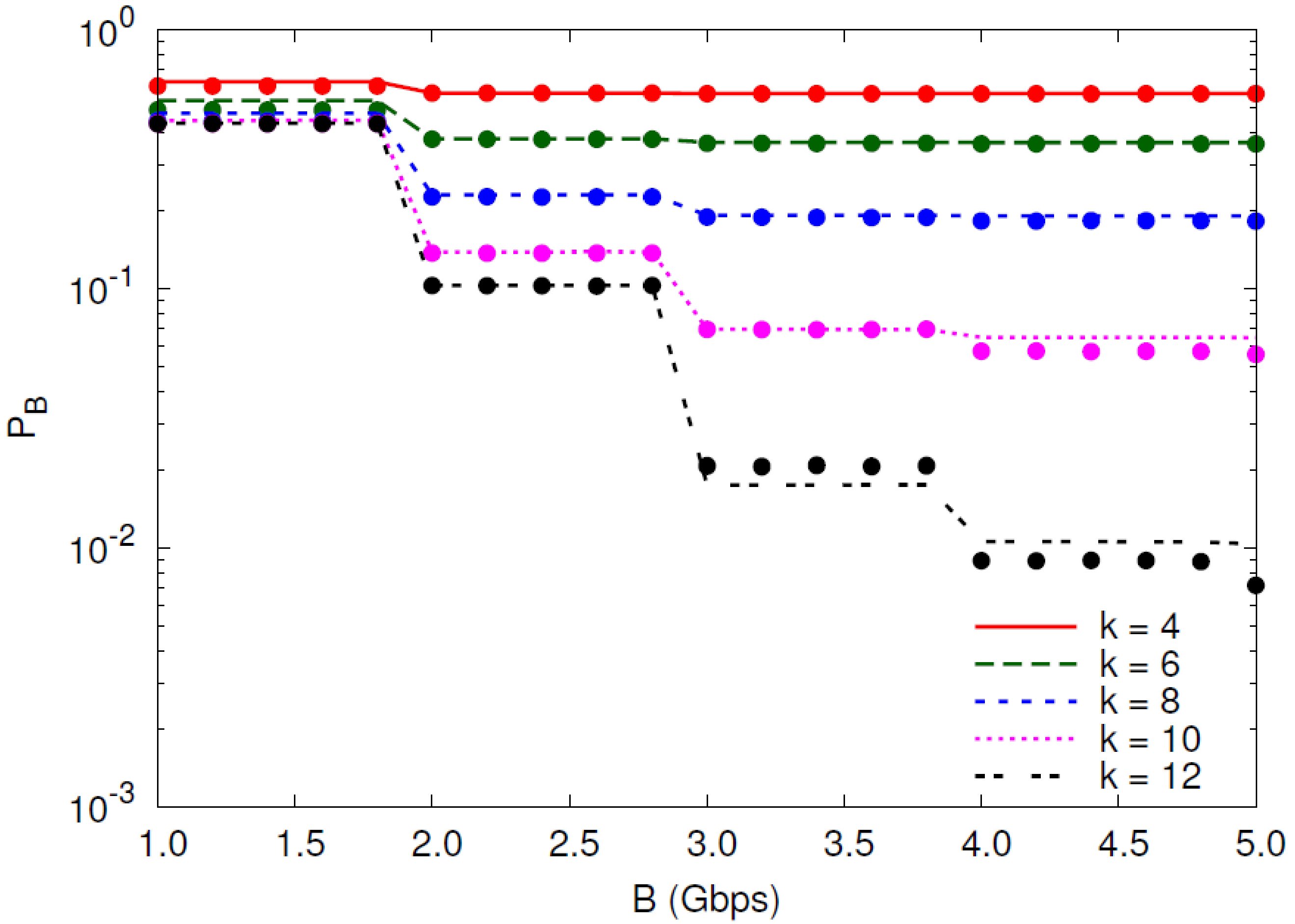
Impact of Computing Resource Sharing



$$B = \infty$$

- Model allows to dimension the cloud federation computing capacity

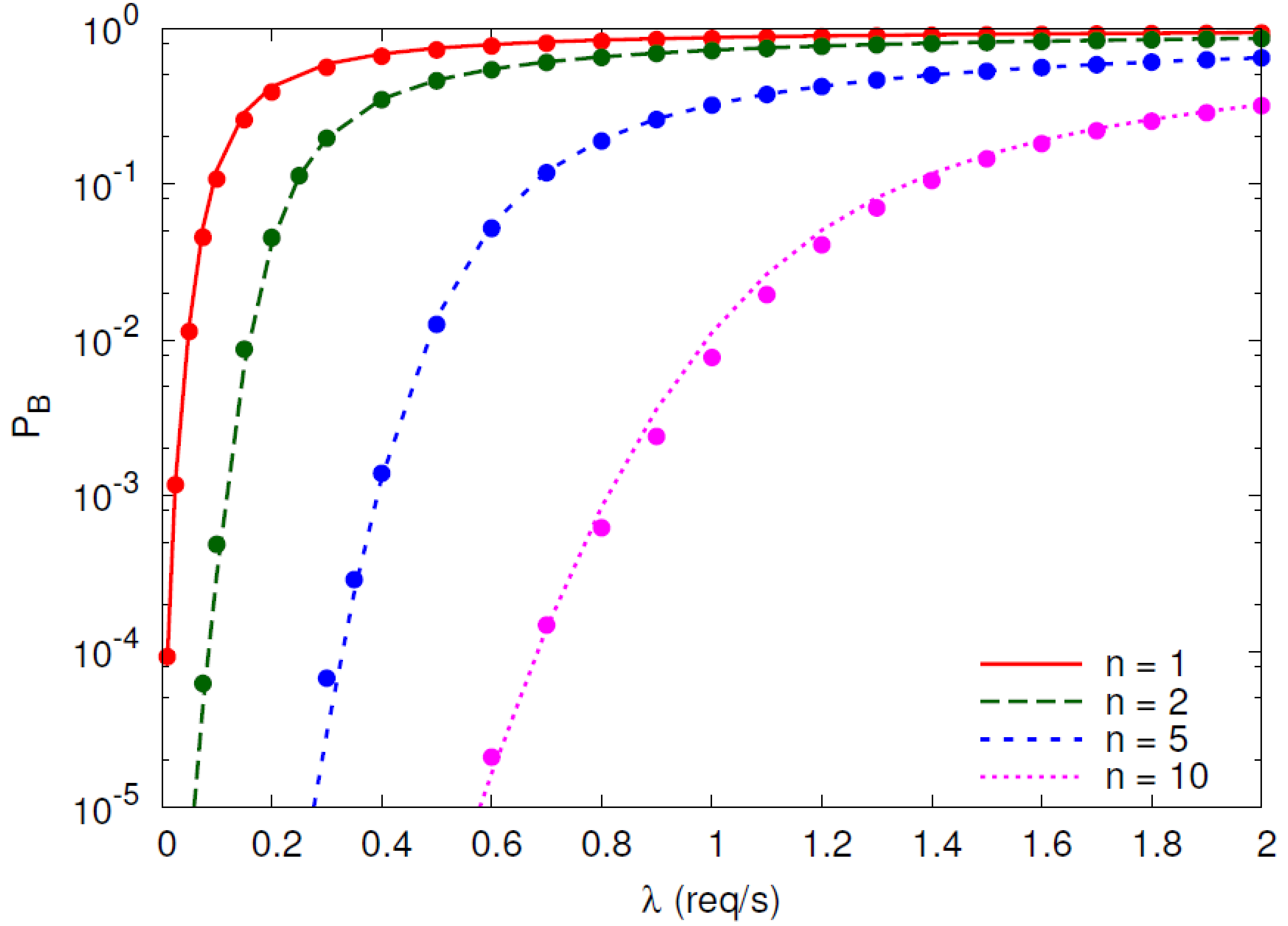
Joint Effect of Netw. and Comp. Resources



$\lambda = 0.8\,\mathrm{req/s}$

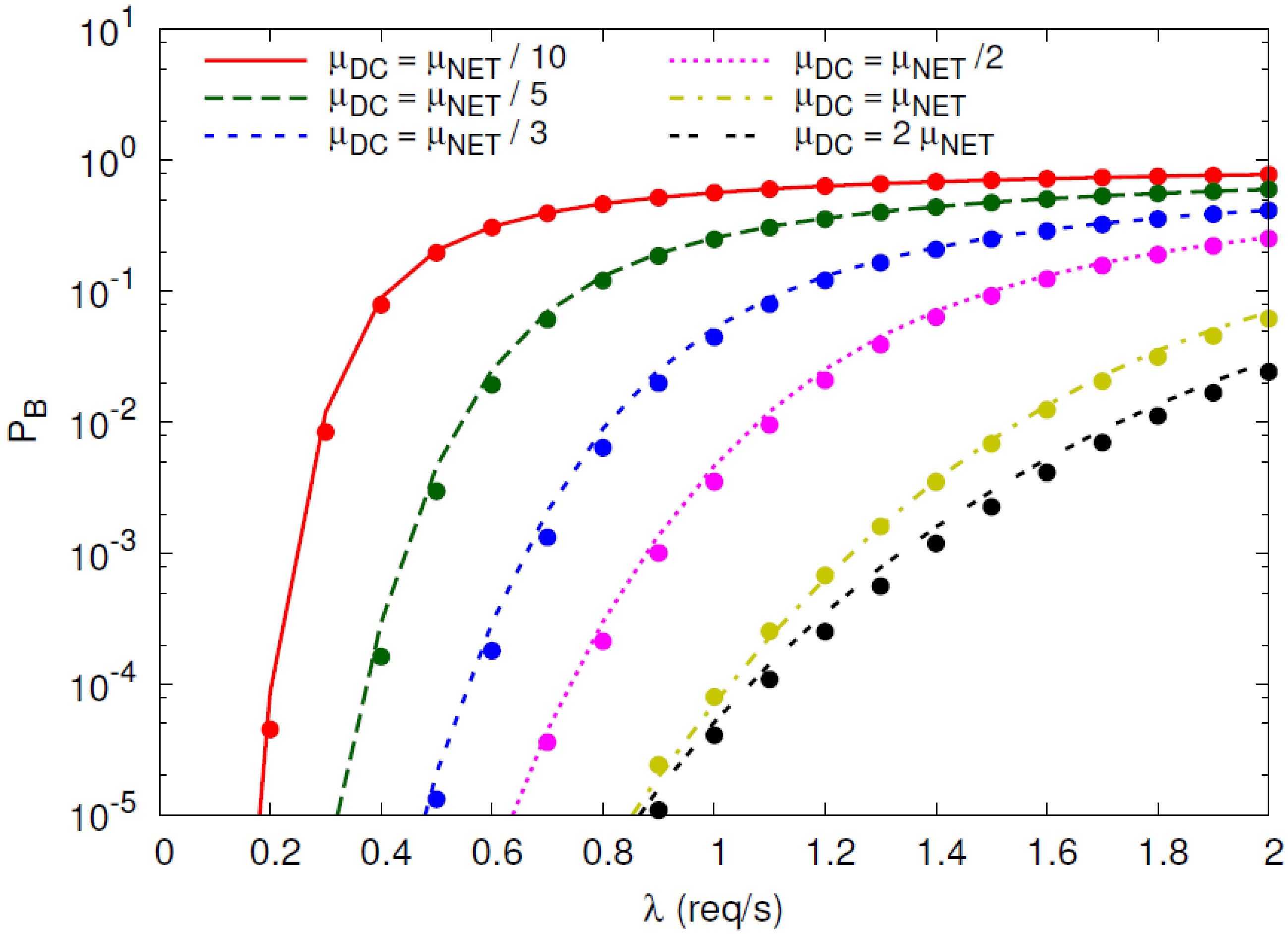
- When *k* is small, lack of computing resources is dominant
- When *k* increases, available bandwidth becomes relevant

Impact of Cloud Federation Size



- Blocking rate can be reduced by increasing the number of DCs
- Need to asses the resulting network infrastructure cost

Impact of Comp. Resource Renewal Rate



- Increasing the renewal rate (e.g., via server consolidation, load balancing, local migration) helps, until network capacity dominates

UNIVERSITÀ DI BOLOGNA
DEISNET

Conclusion

- Analytical model for inter-DC network and shared DC computing resource dimensioning in federated cloud systems

- Network load generated by VM live migration
  - impact on network capacity and computing resource availability
  - trade off network resource usage with end-user’s perceived quality

- Further study on-going
  - release some simplifying assumptions
  - take into account multiple VM migration with different bandwidth allocation stategies
  - consider real DC traffic profiles and VM memory profiles
  - include traffic due to storage migration/synchronization