Case Study #3:
Analysis of Replicated Data with Repair Dependency

Ing-Ray Chen and Ding-Chau Wang

The Computer Journal

Vol. 39, No. 9, 1996, pp. 767-779

Replicated data management

- **Extend Case Study 1 by considering both node and link failures/recovery as well as the effect of repair dependency which occurs when many sites and links may have to share the same repairman due to repair constraints.**

Dynamic voting for replicated data management

- Dynamic voting: Each site S_i *maintains* (VN_i, SC_i, DS_i) to understand if it is in the major partition
- Site *i* is in the major partition if:
  - the number of copies it can access is larger than one half of SC_i
  - the number of copies it can access is exactly equal to one half of SC_i and it can access the "distinguished site" indicated in DS_i
- If a site is in the major partition, it can update locally. After an update is done, all copies in the major partition are updated with the new (VN_i, SC_i, DS_i) value

System model

- Sites and links have independent failure rates λ_s and λ_l.
- A repairman can repair a failed site with rate μ_s and a failed link with rate μ_l.
- There is always an update (called an immediate update) after a failure or repair event since the update rate is much faster than the failure/repair rate

- Site subnet
  - A site can be in one of four states
    - up and current (upcc)
    - up and out-of-date (upoc)
    - down and current (downcc)
    - down and out-of-date (downoc)

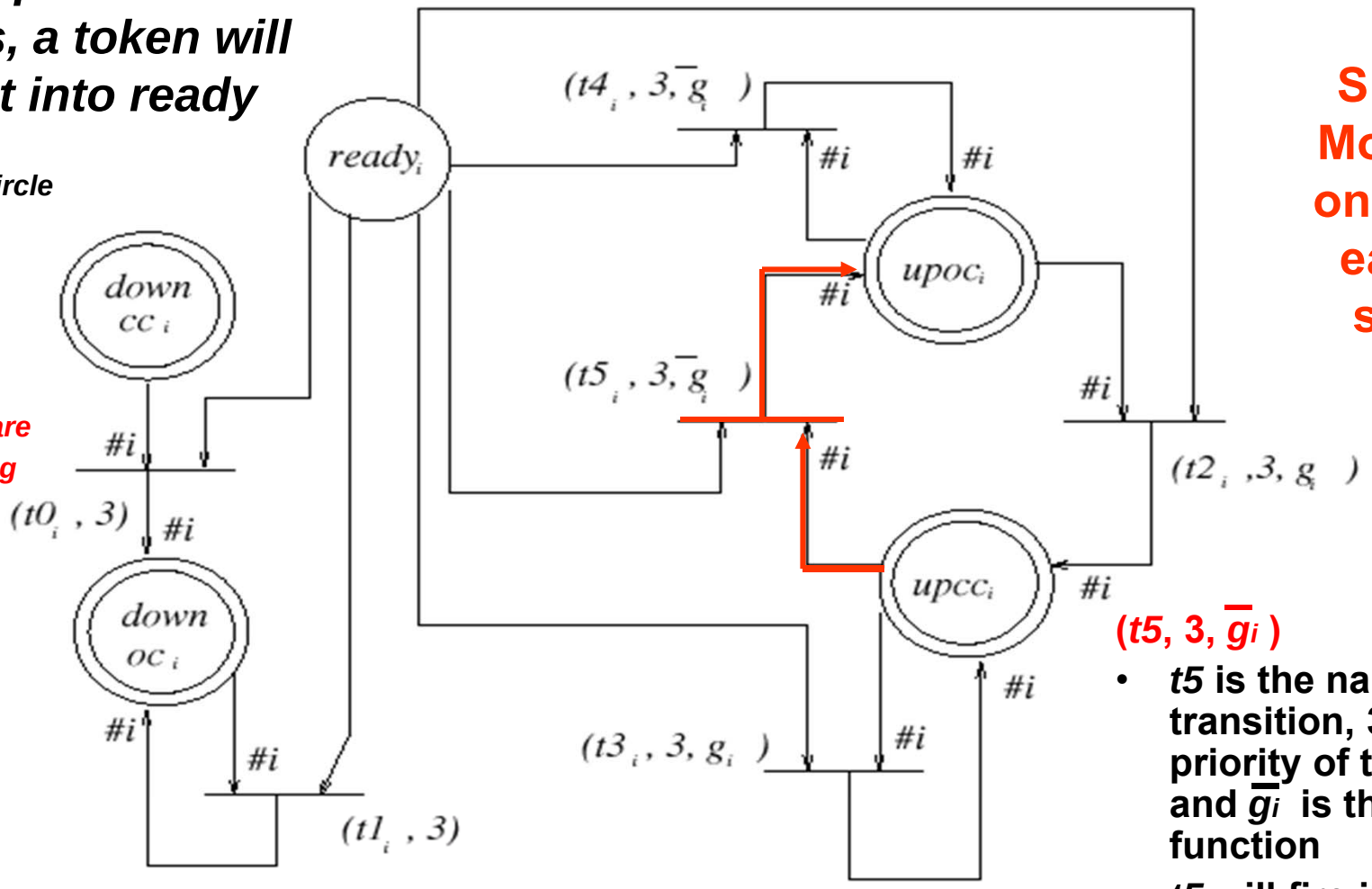
Vhen an update rrives and the ıajor partition xists, a token will e put into ready

g_i true: site *i* is in the major partition

$\bar{g}_i$ true: site *i* is not in the major partition

ɔuble-circle eans a ɔmmon ace

okens are rculating ɔr te i odel



Site *i* Model: one for each site

(*t5*, 3, $\bar{g}_i$)

- ***t5* is the name of the transition, 3 is the priority of the transition, and $\bar{g}_i$ is the enabling function**
- ***t5* will fire if $\bar{g}_i$ is true and site *i* is up and current**
- **After *t5* fires, the state will go from upcc to upoc meaning that site *i* is up and out of date**

)nly one out of the six transitions (t0,t1,t2,t3,t4,t5) an fire, all with the same priority level (3)

FIGURE 1. Local status update actions by site *i*.

g_i true: site *i* is in the major partition

$\bar{g}_i$ true: site *i* is not in the major partition

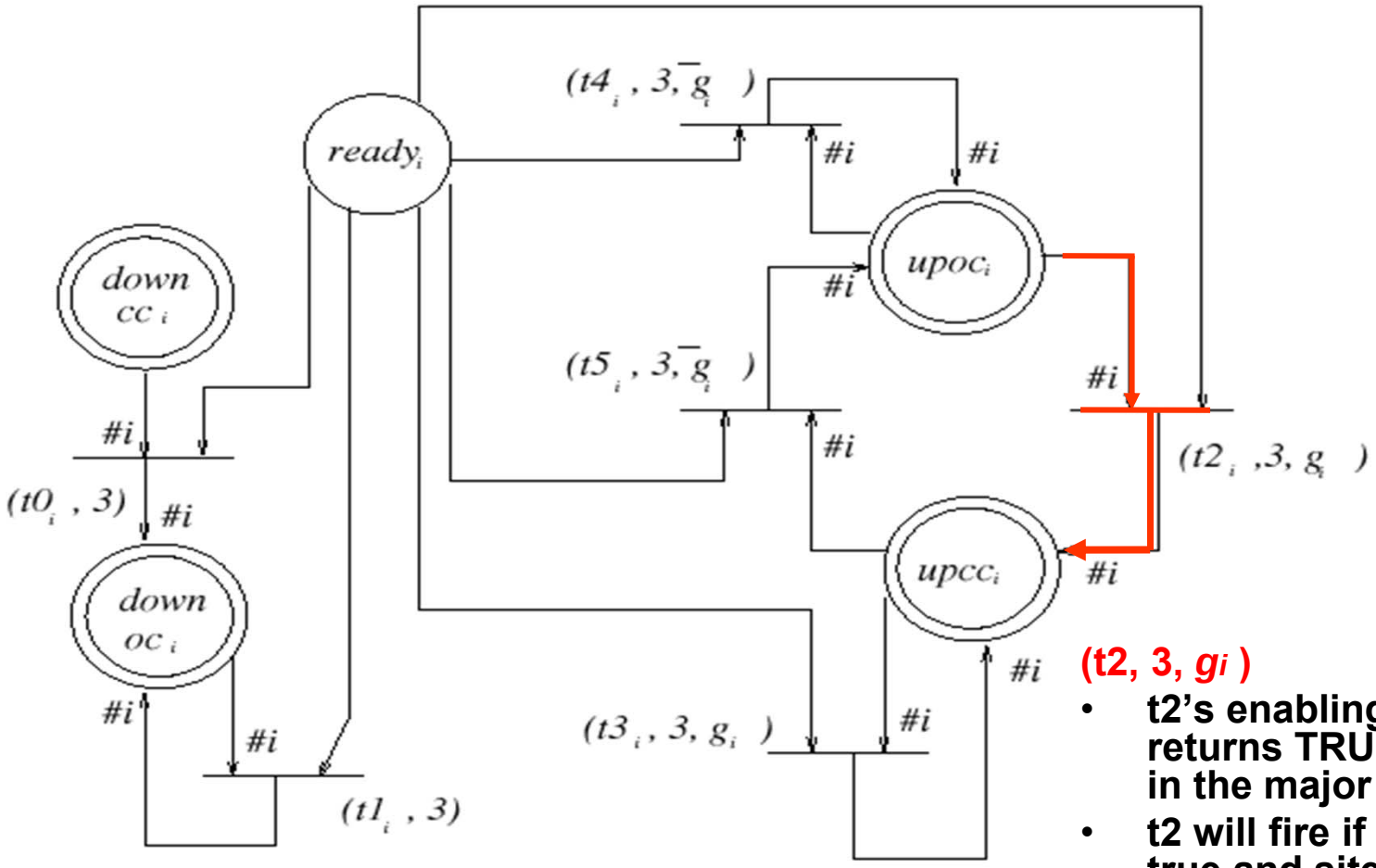


FIGURE 1. Local status update actions by site *i*.

(t2, 3, g_i)

- **t2's enabling function g_i returns TRUE if site *i* is in the major partition**
- **t2 will fire if g_i returns true and site *i* is up and out-of-date**
- **After t2 fires, the state will go from upoc to upcc meaning that the new state will be up and current**

(t0, 3)

- **t0 will fire if site *i* is down and current**
- **After t0 fires, site *i* is down and out of date (downoc)**

g_i true: site *i* is in the major partition

$\bar{g}_i$ true: site *i* is not in the major partition

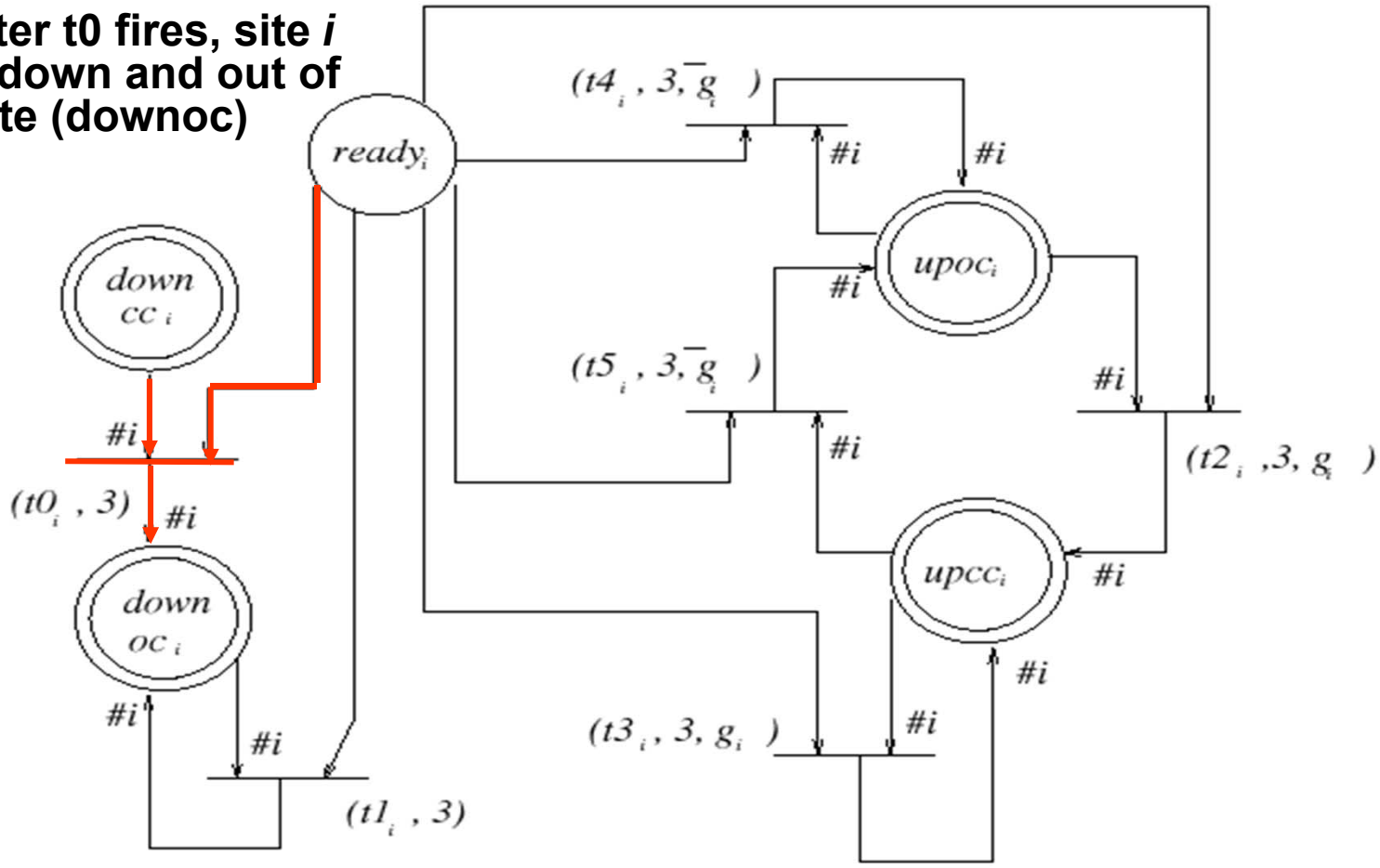
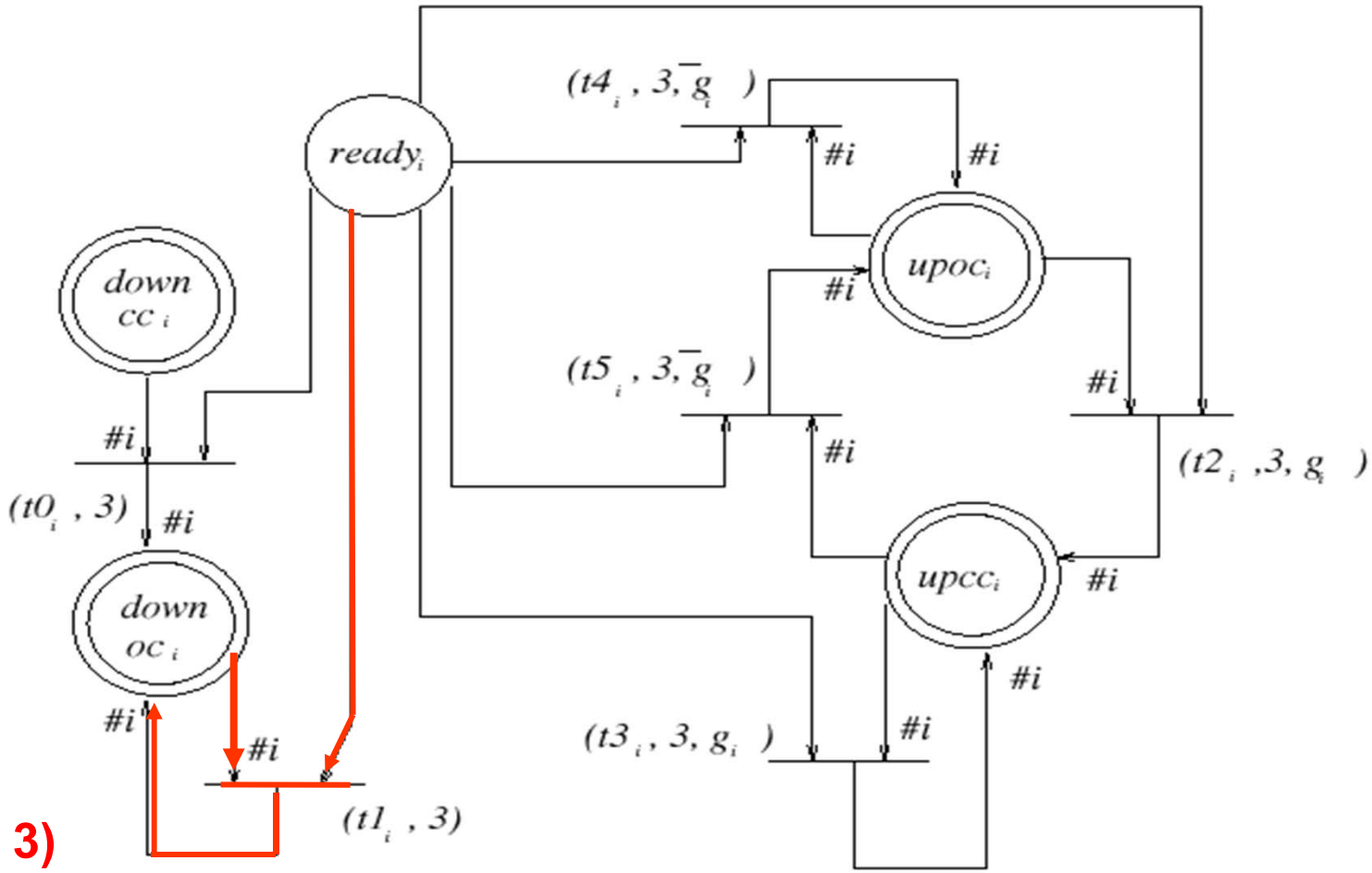


FIGURE 1. Local status update actions by site *i*.

g_i true: site *i* is in the major partition

$\bar{g}_i$ true: site *i* is not in major partition



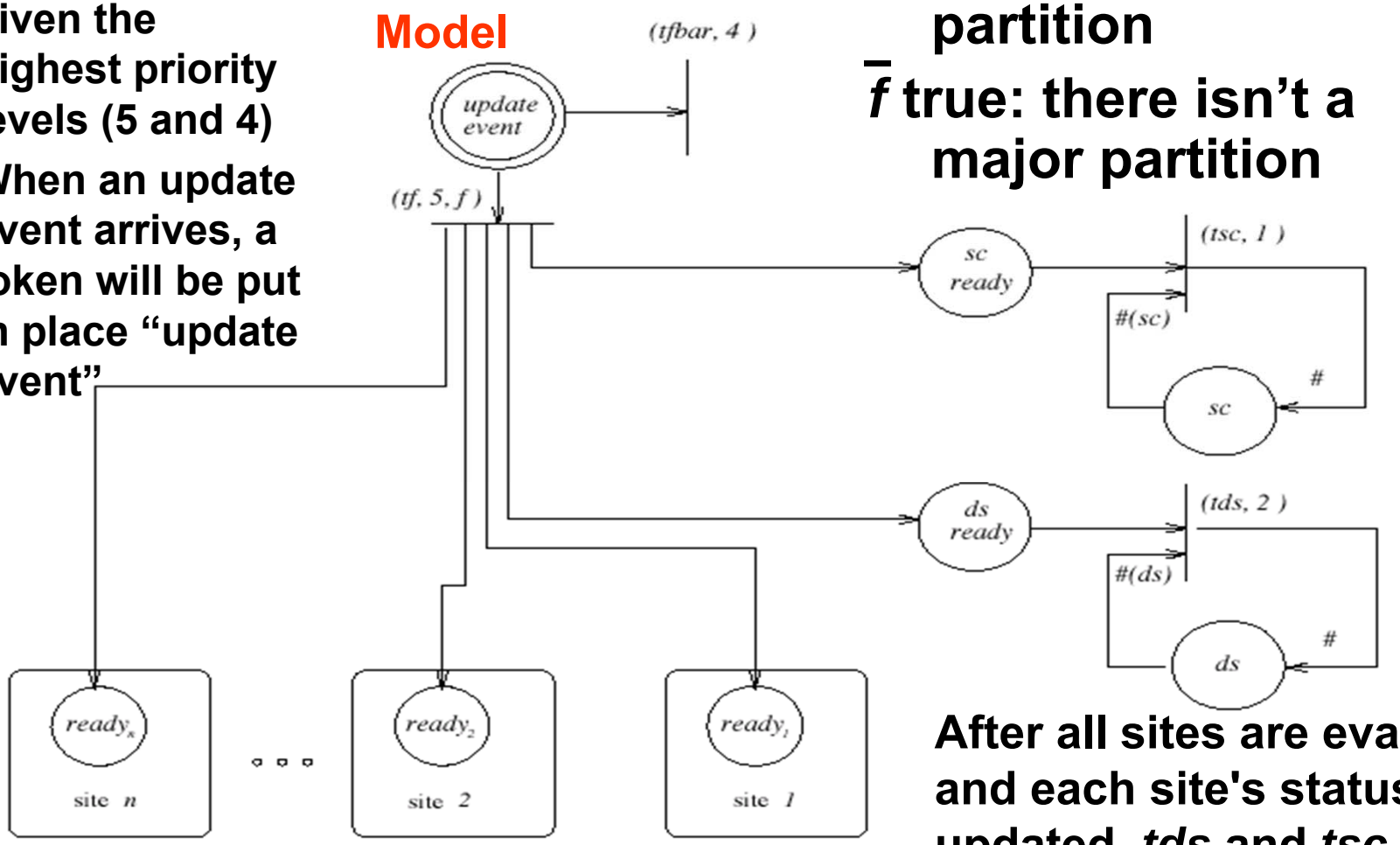
(t1, 3)

- **t1 will fire if site *i* is down and out-of-date**
- **After t1 fires, site *i* remains down and out of date**

FIGURE 1. Local status update actions by site *i*.

- Transitions *tf* and *tfbar* are given the highest priority levels (5 and 4)
- When an update event arrives, a token will be put in place "update event"

System Subnet Model



Each of the boxes labeled site *i* is the site subnet model for site *i*

***f* true: there is a major partition**

$\bar{f}$ true: there isn't a major partition

After all sites are evaluated and each site's status are updated, *tds* and *tsc* which have lowest priority levels will execute

FIGURE 2. Global status-update actions triggered by an update operation.

(tsc, 1) updates the site cardinality as the number of tokens in place sc:

- **Input arc multiplicity: #(sc)**
- **Output arc multiplicity: the number of sites with mark (upcc) >0 (i.e., in the major partition)**

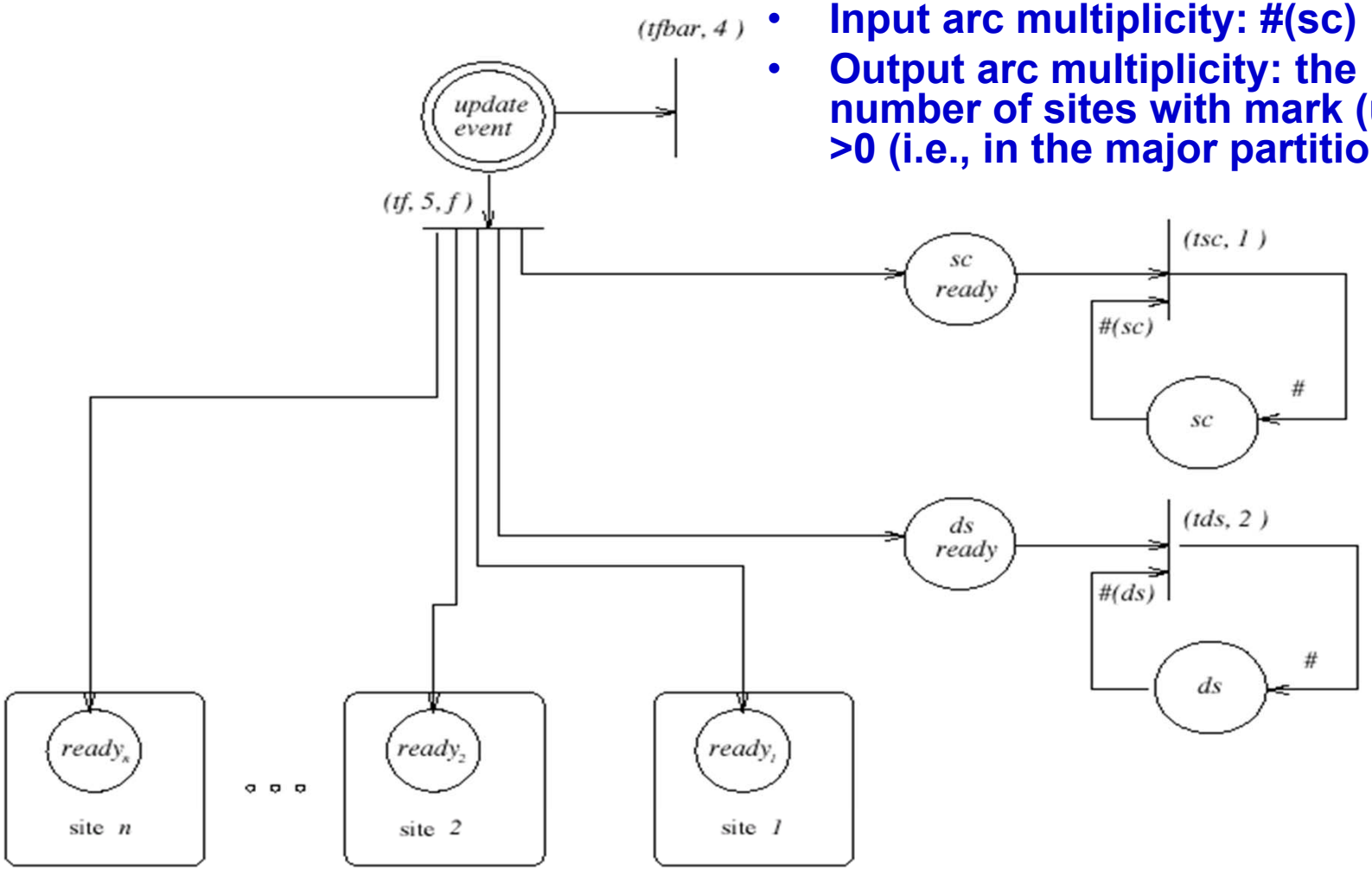
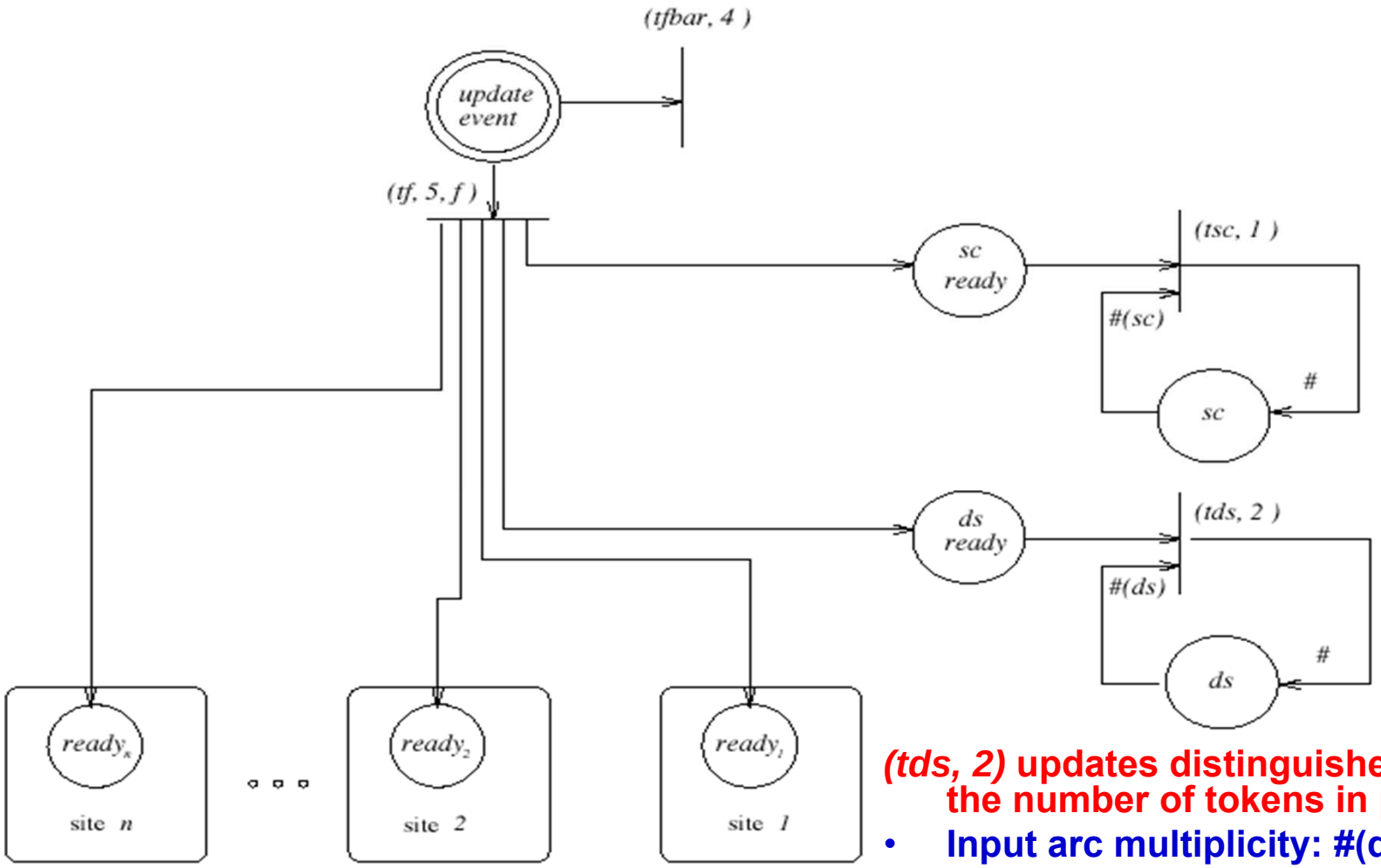


FIGURE 2. Global status-update actions triggered by an update operation.



***(tds, 2)* updates distinguished site as the number of tokens in place ds:**

- **Input arc multiplicity: #(ds)**
- **Output arc multiplicity: maximum #(upcc) among all sites in the major partition**

FIGURE 2. Global status-update actions triggered by an update operation.

Independent Repairman Model

- This subnet describes the effect of site *i*'s failure and repair on the system state
- site *i* can only be in one state at a time, so only one transition out of these two subnets is possible at any time.

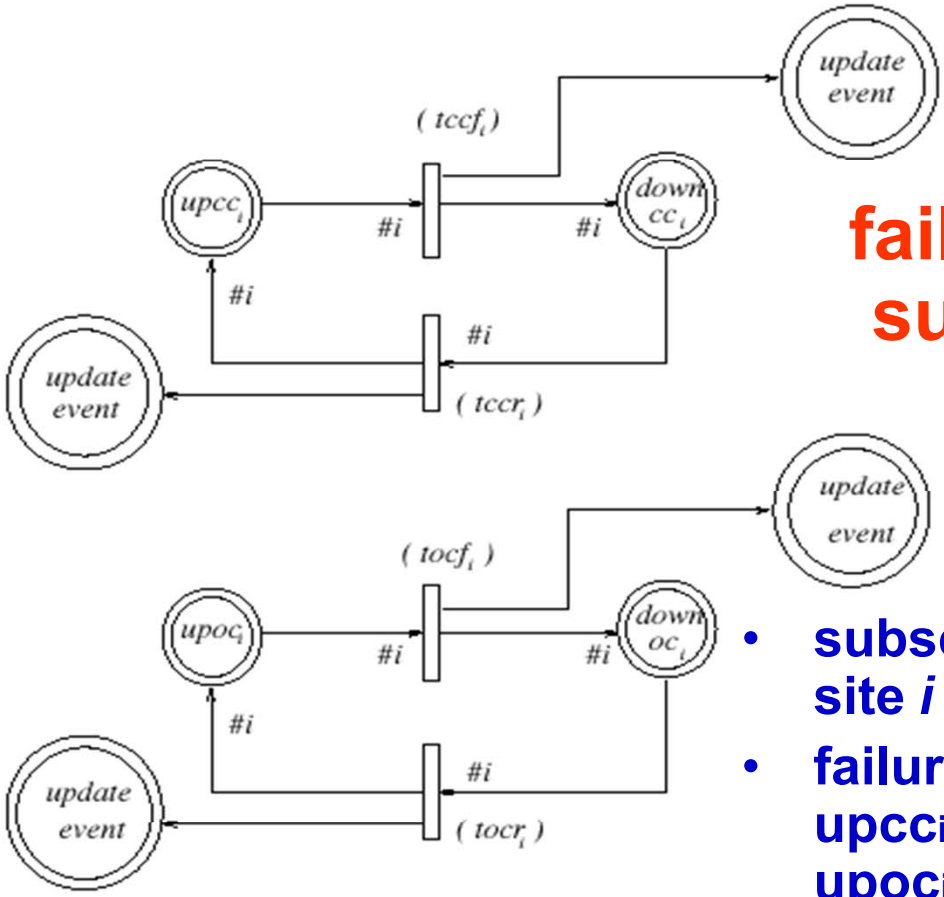


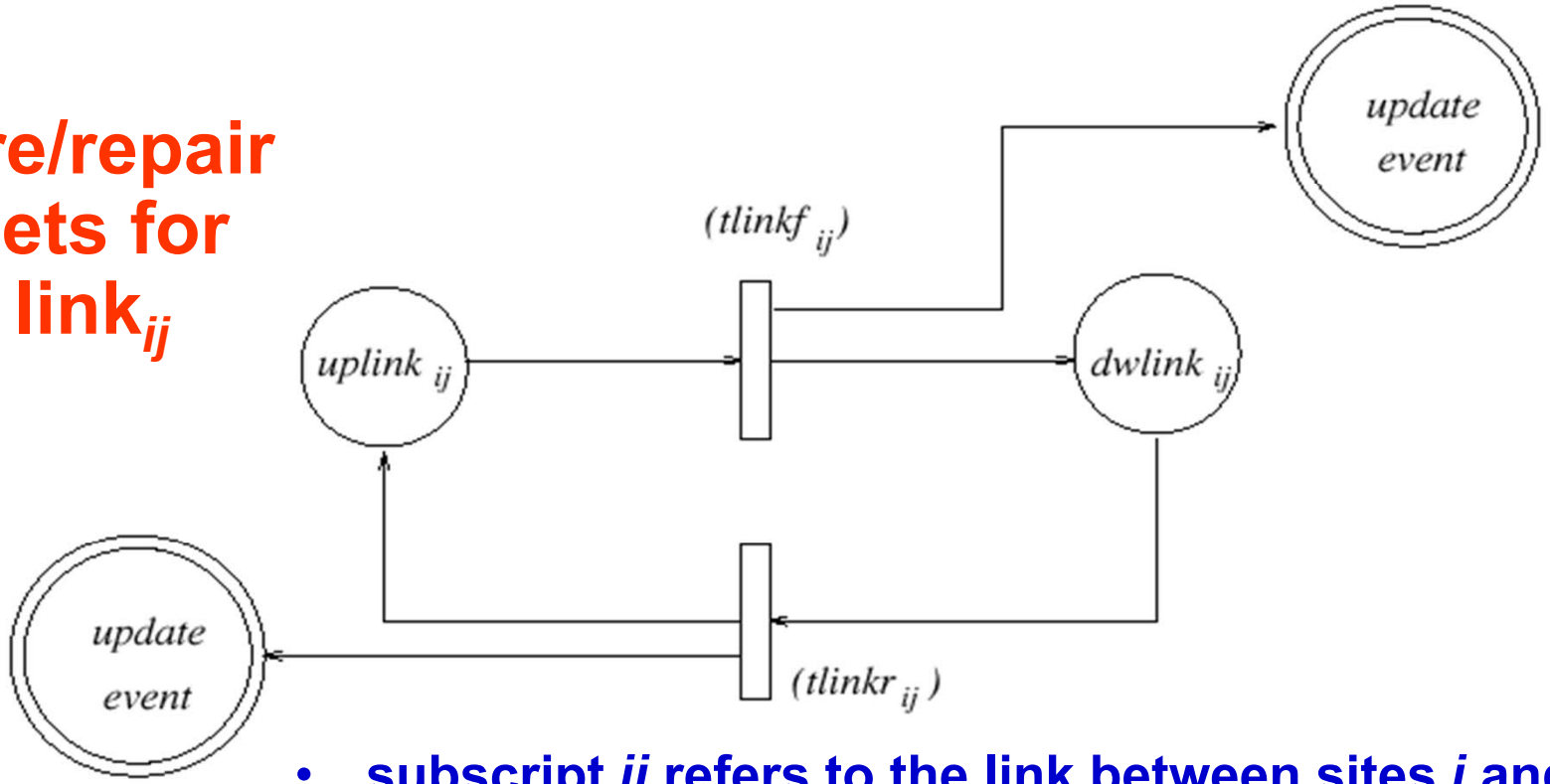
FIGURE 3. Site failure/repair events.

Site failure/repair subnets for site *i*

- subscript *i* refers to site *i*
- failure events: upcc$_i$ → dwcc$_i$ and upoc$_i$ → dwoc$_i$ with rate of λ_s
- repair events: dwcc$_i$ → upcc$_i$ and dwoc$_i$ → upoc$_i$ with rate of μ_s

Independent Repairman Model

Link failure/repair subnets for each link$_{ij}$



- subscript *ij* refers to the link between sites *i* and *j*
- failure events: uplink$_{ij}$ → dwlink$_{ij}$
  - With rate of λ_l
- repair events: dwlink$_{ij}$ → uplink$_{ij}$
  - With rate of μ_l

FIGURE 4. Link failure/repair events.

TABLE 1. Meanings of places.

Place	Meaning
$upcc_i$	Copy_i is up and current
$downcc_i$	Copy_i is down and current
$upoc_i$	Copy_i is up and out of date
$downoc_i$	Copy_i is down and out of date
$uplink_{ij}$	Link_{ij} is up
$dwlink_{ij}$	Link_{ij} is down
$update\_event$	An update is initiated
sc_ready	An SC is initiated
sc	$\#(sc)$ indicates the SC
ds_ready	A DS change is initiated
ds	$\#(ds)$ indicates the ID of the DS
$ready_i$	A local update at site i is in process

TABLE 2. Arc multiplicity functions.

Arc	Multiplicity
$sc \rightarrow tsc$	$\#(sc)$
$tsc \rightarrow sc$	# of sites in the major partition with mark($upcc$) > 0
$ds \rightarrow tds$	$\#(ds)$
$tds \rightarrow ds$	Max $\#(upcc)$ among all sites in the major partition

TABLE 4. Enabling functions.

Tr.	Enabling function
tf	$f()$ {IF $\exists$ a partition $\mathcal{M}$ with sum equal to # of sites in $\mathcal{M}$ with mark($upcc$) > 0; AND IF (sum $>$ #(sc)/2) OR (sum = #(sc)/2 AND mark($upcc_{\#(ds)}$) AND site #(ds) $\in \mathcal{M}$) THEN RETURN 1; ELSE RETURN 0}
$t2_i$	$g_i()$ {Look at mark($uplink_{jk}$) $\forall j \forall k$ to determine site i's partition; IF site i's in the major partition $\mathcal{M}$ THEN RETURN 1; ELSE RETURN 0}
$t3_i$	$g_i()$
$t4_i$	$\bar{g}_i()\{1 - g_i()\}$
$t5_i$	$\bar{g}_i()$

FIFO repairman model (one repairman)

- We can make use of the independent repairman model and modify the repair rates to account for repair dependency.
- The repair rate is “deflated” by the total number of failed sites and links to account for the effect of repair resource sharing
- If a state has 3 failed entities: two failed sites and one failed link,
  - For the independent repairman model, repair rates are μ_s, μ_s and μ_l
  - For the FIFO repairman model, repair rates are $\mu_s / 3$, $\mu_s / 3$ and $\mu_l / 3$.

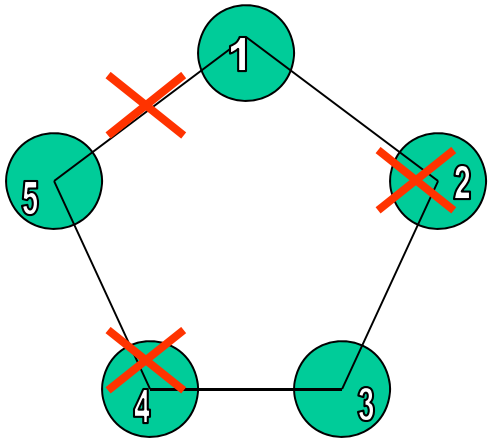
TABLE 5. Rates of timed transitions for FIFO repair.

Timed Tr.	Rate value
$tccf_i$	λ_s
$tccr_i$	$\dfrac{\mu_s}{\sum_{j,k,k\neq j} \#(downcc_j + downoc_j + dwlink_{jk})}$
$tocf_i$	λ_s
$tocr_i$	$\dfrac{\mu_s}{\sum_{j,k,k\neq j} \#(downcc_j + downoc_j + dwlink_{jk})}$
$tlinkf_{ij}$	λ_l
$tlinkr_{ij}$	$\dfrac{\mu_l}{\sum_{j,k,k\neq j} \#(downcc_j + downoc_j + dwlink_{jk})}$

Linear-order repairman model (one repairman)

- **Repairing failed site/link in a prescribed order**
- **Creating a new enabling function associated with each repair transition**
- **Only one enabling function at any state returns TRUE based on the prescribed linear order and all others return FALSE**

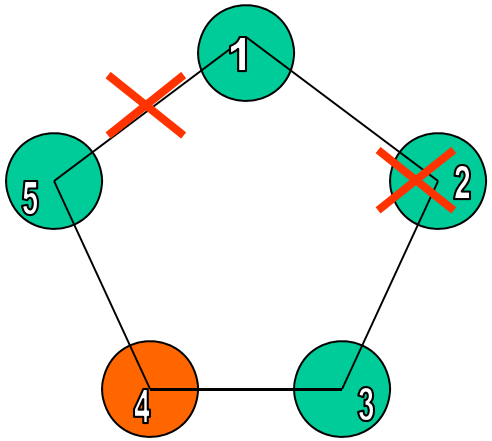
Linear-order repairman model



- A 5-site ring topology with the linear repair order being sites 5,4,3,2,1 followed by links 45,51,43,32,21

- If sites 4 and 2, and link 51 are down, then site 4 is chosen to be repaired first

Linear-order repairman model



- Enabling functions associated with sites 4 and 2 and link 51 will return TRUE, FALSE and FALSE, respectively, meaning that site 4 will be repaired first over site 2 and link 51.

TABLE 6. Enabling functions for linear-order repair.

Transition	Enabling function
$tccr_i, tocr_i$	$h1_{site}(i)$ {IF site i failed and the repair rank of site i is higher than those of other failed sites or links in the linear order THEN RETURN TRUE; ELSE RETURN FALSE}
$tlinkr_{ij}$	$h1_{link}(i, j)$ {IF link ij failed and the repair rank of link ij is higher than those of other failed sites or links in the linear order THEN RETURN TRUE; ELSE RETURN FALSE}

Best-first repairman model (one repairman)

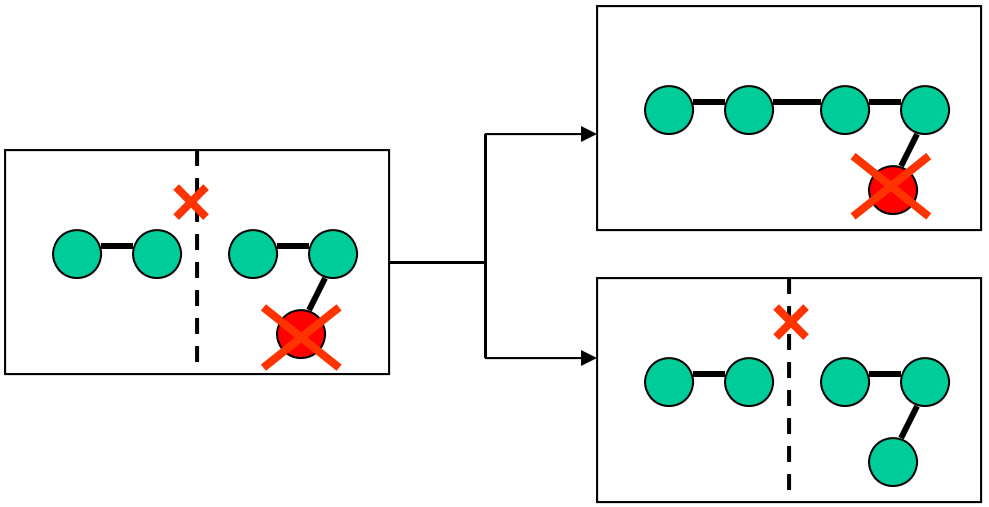
- Preference is given to a failed site or link whose repair can lead to the existence of a major partition with respect to the current state

- If there are more than one failed sites or links whose repair would lead to the existence of a major partition, then a tie-breaker rule will be applied to select one to be repaired next.

Best-First Repair Strategy
Tie-Breaker Rules

- Choosing a failed entity such that after repair it will result in more current copies (i.e., a large **SC**) in the major partition (i.e., the more upcc sites in the major partition, the better)
- Choosing a site (among failed sites) with the highest linearly ordered site ID, so it has a higher chance to become the **DS**
- Choosing a failed entity that will stay alive for a longer time after repair. That is, choose one with a lower failure rate and a higher repair rate. For example, when choosing between a failed site vs. a failed link, if $\mu_s / \lambda_s > \mu_l / \lambda_l$, then repair the failed site, otherwise repair the failed link

Best-First Repair Example



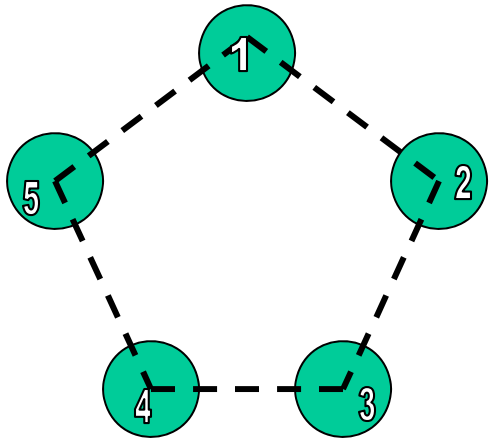
Effective:
(SC is 4 after repair)

Ineffective:
(SC is 3 after repair)

TABLE 7. Enabling functions for best-first repair.

Transition	Enabling function
$tccr_i, tocr_i$	$h1_{site}(i)$ {IF site i failed and the hypothetical site availability after repairing site i is higher than those of other failed sites or links in the system THEN RETURN TRUE; ELSE RETURN FALSE}
$tlinkr_{ij}$	$h1_{link}(i, j)$ {IF link ij failed and the hypothetical site availability after repairing link ij is higher than those of other failed sites or links in the system THEN RETURN TRUE; ELSE RETURN FALSE}

Evaluation



- Tested with a 5-site ring topology
- Four repairman models:
  - Independent repair
  - Dependent repair (one repairman)
    - FIFO
    - Linear-order
    - Best-first

Model complexity: number of states

	Independent	FIFO	Linear-order	Best-first
# of states in the underlying Markov model	8674	8674	5429	3821

Performance metrics and reward assignments for calculation

	Definition	Reward Assignment
System Availability	The steady-state probability that a major partition exists	Reward rate = 1 for those states in which enabling function *f*() is evaluated to TRUE. Reward rate = 0, otherwise
Site Availability	The probability that an update arriving at an arbitrary site will succeed	Reward rate = 1*k/n for those states in which enabling function *f*() is evaluated to TRUE where k is the number of up and current copies in the major partition. Reward rate = 0, otherwise
k: # of 'up and current' (upcc) sites in the major partition in a particular state n: total number of sites in a system (n=5 in a 5-site ring topology)		

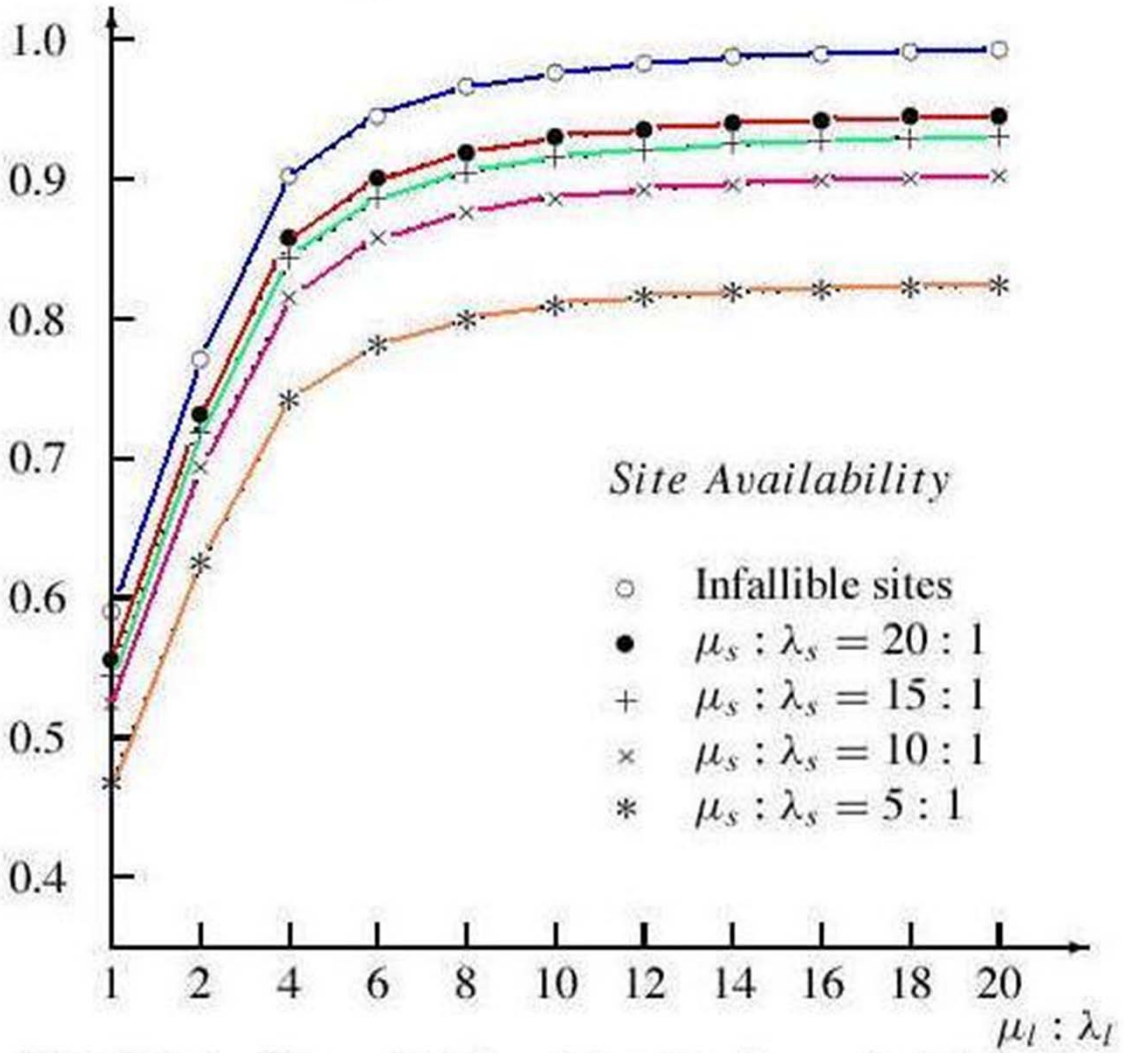


FIGURE 5. Site availability of five-site ring under independent repairman model.

Results:

independent repairman model

μ_l:λ_l↑ site availability↑

μ_s:λ_s↑ site availability↑

Site failure only assumption (as in Case Study 1) will **overestimate** the site availability unrealistically.

Results: Comparison of repairman models

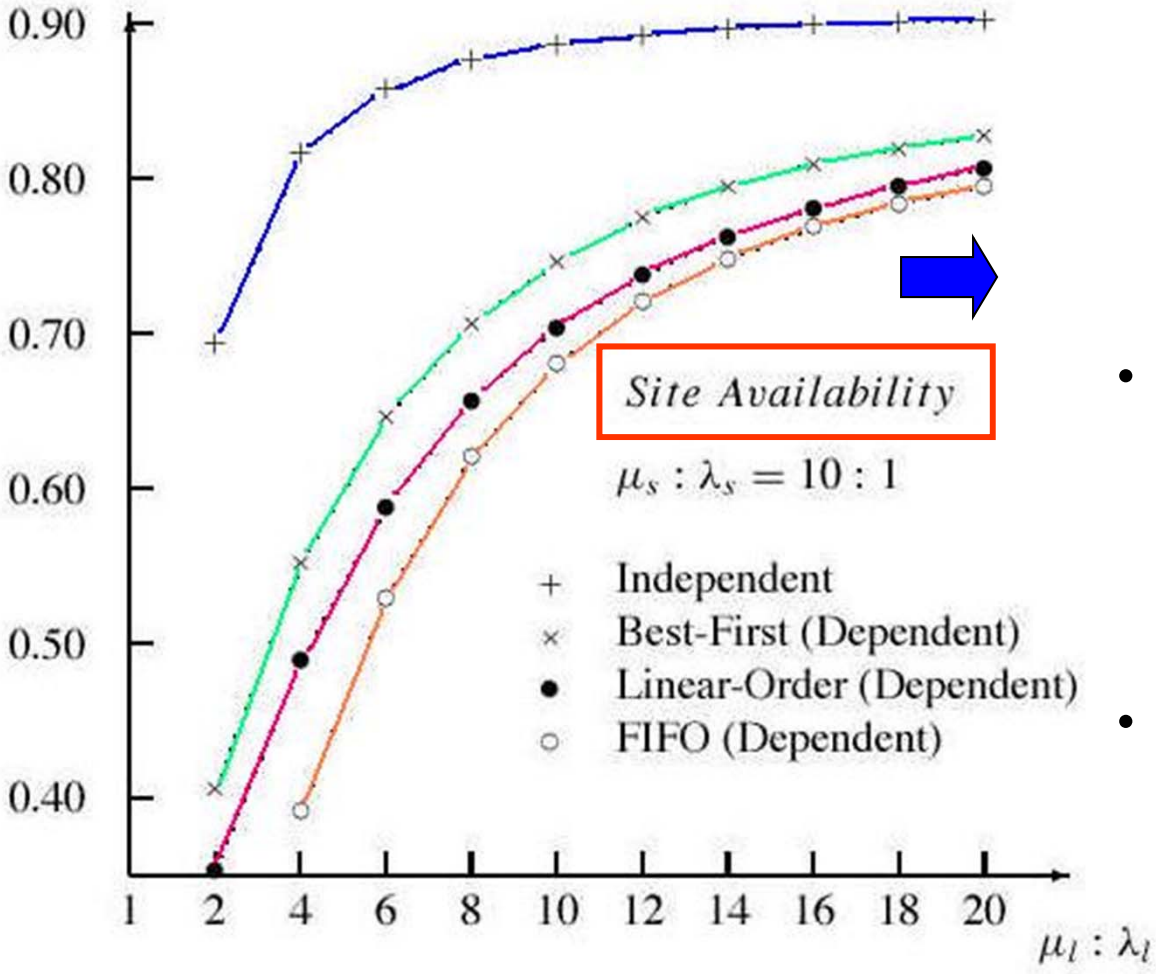


FIGURE 6. Site availability of five-site ring under various repairman models.

- Site availability under independent repair is much higher than that under dependent repair
- Among dependent repair:
  - Best-first > Linear-Order > FIFO