Dependable Systems

State-Based Dependability Modeling

Dr. Peter Tröger

Sources:

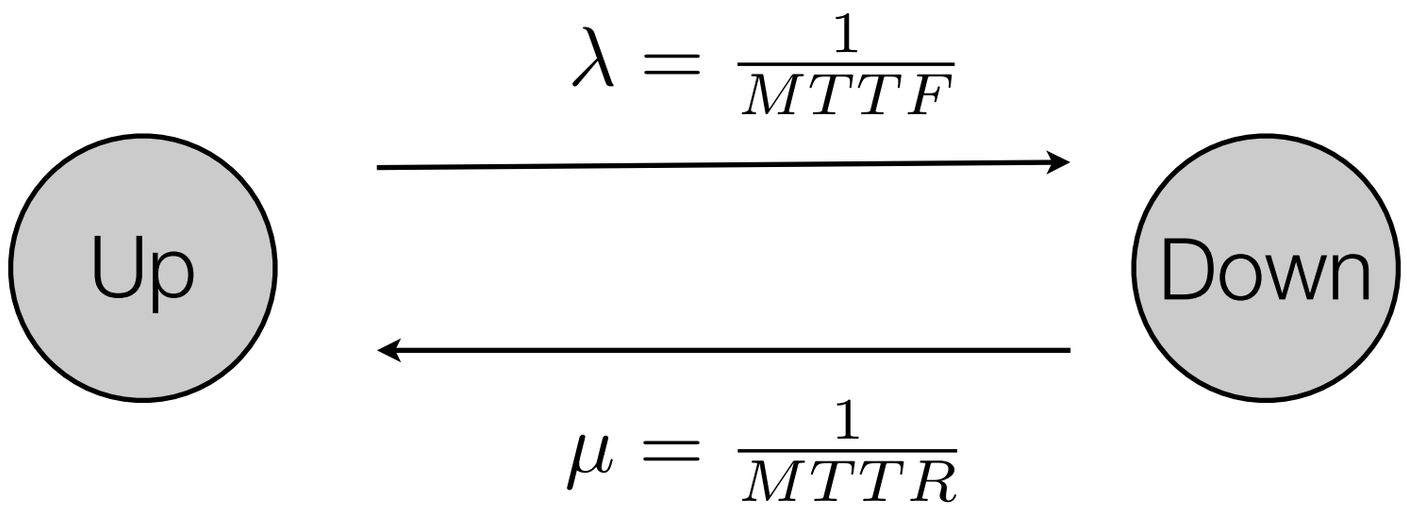
Eusgeld, Irene et al.: Dependability Metrics. 4909. Springer Publishing, 2008

Menasce, Daniel A.; Almeida, Virgilio A.: Capacity Planning for Web Services: Metrics, Models, and Methods. Prentice Hall, 2002. , 0-13-065903-7

Dependability Modeling

- Default approach: Utilize a formalism to model system dependability
  - Quantify the availability of components, calculate system availability based on this data and a set of assumptions - the availability model
    - Most models expose the same expressiveness
    - Each formalism allows to focus on certain aspects
    - Component-based models: Reliability block diagram, fault tree
    - State-based models: Markov chains, petri nets
- System understanding evolved from hardware to software to IT infrastructures
  - Example: Organization management influence on business service reliability
    - Information Technology Infrastructure Library (ITIL)
    - CoBiT(Control Objectives for Information and related Technology)

Dependability Modeling



- Some assumptions
  - All failure and repair events are exponentially distributed
  - Components are either fully working or completely failed
  - All failure and repair events are pair-wisely stochastically independent
  - Correct functioning at t can be treated as event, with event probability derived from the availability value computed by failure rate and repair rate

State-based Models

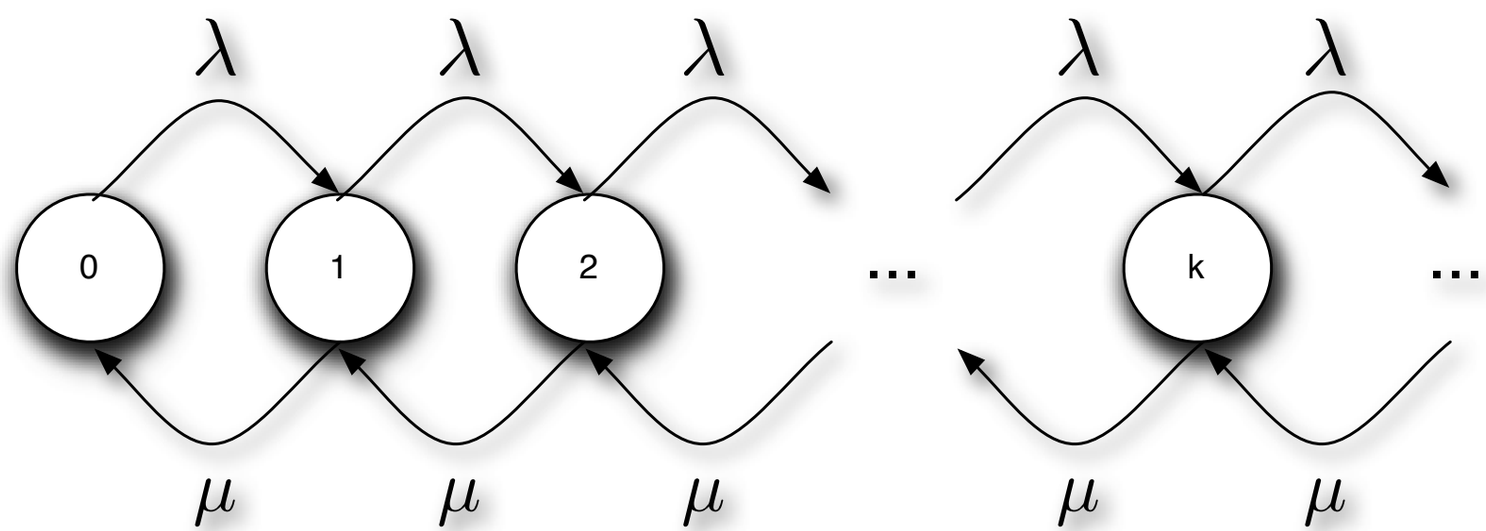
- Component-based models work well if failure events are stochastically independent
  - But: Catastrophic events destroy multiple components
- State-based models focus on failure states of the system
  - Can handle transitions between failure states
  - Analytical solution
    - Demands independent failures, constant failure rates, (exponential distribution)
  - Solution through simulation
    - State model is simulated to estimate the resulting dependability metrics
    - Arbitrary failure event distributions, approximations, long simulation time

State Transition Diagrams

- Modeling approach, typically used for queueing systems
- Basic assumptions
  - *Homogeneous workload assumption* - All request are indistinguishable, so only their sum counts
  - *Operational equilibrium* - Number of requests in the system is the same at the start and end of investigation
    - May vary in the interval, but average throughput is constant
    - Number of departures tends to approach the number of arrivals - „all forces on the object are balanced“
  - *Memoryless assumption*
    - Server state is a single parameter - number of processed requests

State Transition Diagrams

- Transitions between states happen at some rate
  - Arrival rate λ (transitions / sec), request completion rate μ (transitions / sec)
  - *Flow-In Flow-Out principle*
    - Operational equilibrium ensures that transitions into the state are equal to transitions out of that state

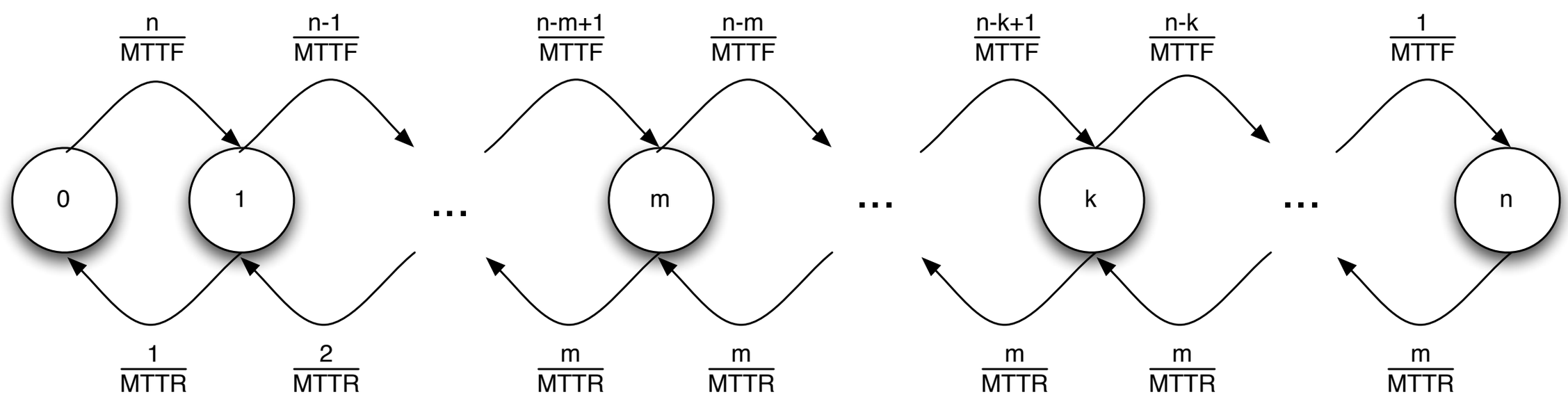


Application of State Transition Diagrams

- System of *n* parallel servers which ‚arrive‘ at repair situation, maximum number *m* of parallel repair activities, maximum *k-out-of-n* servers are allowed to be failed
  - State expresses number of servers down
  - Arrival rate == failure rate
  - Completion rate == repair rate
  - Transitions express a component failure or a component back in operation

$$\lambda_k = \frac{n-k}{MTTF}$$

$$\mu_k = \begin{Bmatrix} k/MTTR & k = 1, ..., m \\ m/MTTR & k = m+1, ..., n \end{Bmatrix}$$



Markov Chains

- Discrete random process, usually drawn as state transition diagram
  - **Markov property** - Next step depends only on the current step

$$P(X_{n+1}|X_1, X_2, \ldots, X_n) = P(X_{n+1}|X_n).$$

  - Impossible to predict future states, but useful for statistical properties
  - Finite state space (**chain**), transitions with probabilities, initial state probabilities
- **Transient state** - Probability to not return to this state (finite number of visits)
- **Recurrent state** - Probability of 1 to return to this state after unspecified time t
  - *Mean recurrence time* can be used as MTTF metric
- **Time-homogeneous Markov chains** - Transition probabilities do not change in time

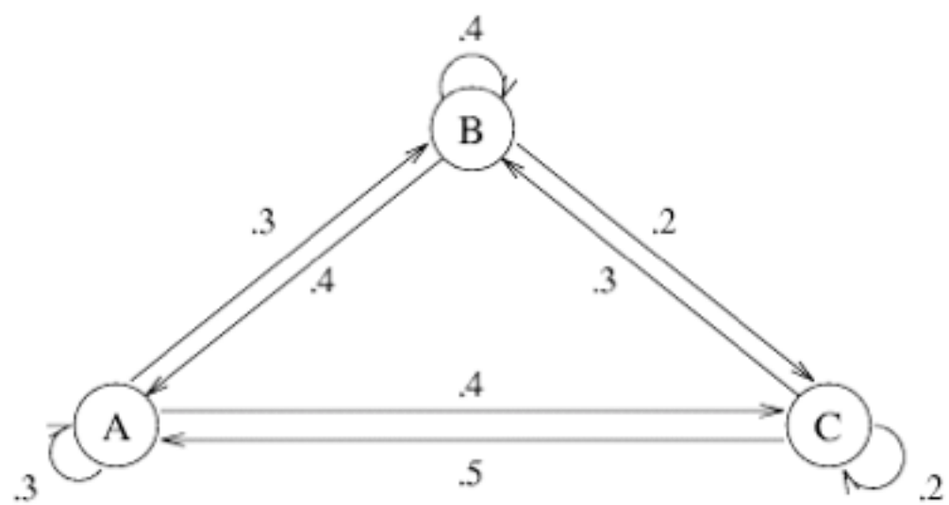
$$\Pr(X_{n+1} = x|X_n = y) = \Pr(X_n = x|X_{n-1} = y)$$

Markov Chains

- **Discrete-time Markov chain (DTMC)**
  - System state only changes after a fixed time interval, system is in exactly one state
  - Transition to next state depends on transition probability (non-negative) at t
  - Each row of the probability transition matrix represents flow out of that state, the columns the transition flow into the state, row sum is one
- **Continuous-time Markov chain (CTMC)**
  - Allows state changes at any instance of time - continous parameter space, still discrete state space
  - Transition to next state after spending some time in a state - *holding time*
    - Generator matrix Q therefore expresses transition rates instead of probabilities
  - By definition, the diagonal entries are equal to minus the total rate out of that state
    - Rates with which no state change takes place

Markov Chains - DTMC Example

$$q_{i,j} = \lim_{\Delta t \to 0} \frac{\mathrm{P}\{X_{t+\Delta t} = j \,|\, X_t = i\}}{\Delta t} \qquad i \neq j$$



$$\begin{array}{c c} & \begin{array}{ccc} \mathbf{A} & \mathbf{B} & \mathbf{C} \end{array} \\ \begin{array}{c} \mathbf{A} \\ \mathbf{B} \\ \mathbf{C} \end{array} & \begin{bmatrix} .3 & .3 & .4 \\ .4 & .4 & .2 \\ .5 & .3 & .2 \end{bmatrix} = S \end{array}$$

Transition Matrix

$$\begin{array}{c c} & \begin{array}{ccc} \mathbf{A} & \mathbf{B} & \mathbf{C} \end{array} \\ \begin{array}{c} \mathbf{A} \\ \mathbf{B} \\ \mathbf{C} \end{array} & \begin{bmatrix} .41 & .33 & .26 \\ .38 & .34 & .28 \\ .37 & .33 & .30 \end{bmatrix} = S^2 \end{array}$$

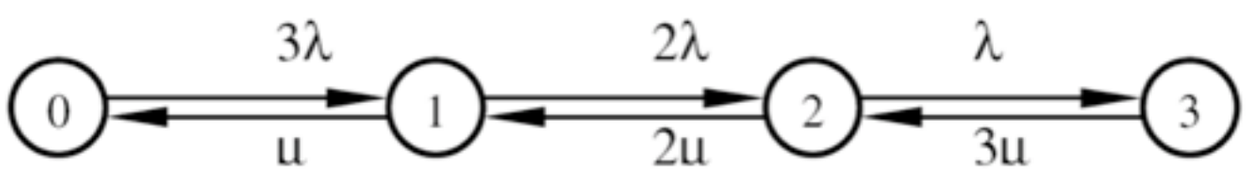
Probability Matrix after 2 steps

- Each row sum of the transition matrix is 1
- Initial distribution vector can be combined with transition matrix to find probabilities for being in one of the states after one step

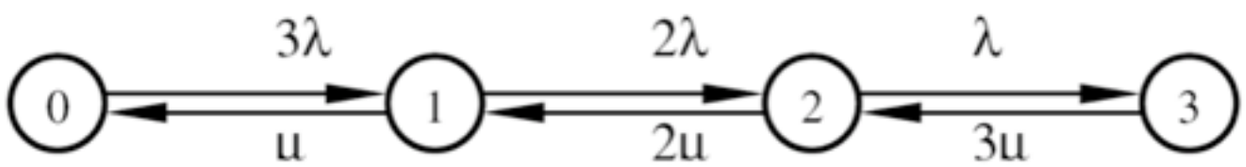
$$\begin{bmatrix} .\overline{3} & .\overline{3} & .\overline{3} \end{bmatrix} \begin{bmatrix} .3 & .3 & .4 \\ .4 & .4 & .2 \\ .5 & .3 & .2 \end{bmatrix} = \begin{bmatrix} .4 & .\overline{3} & .2\overline{6} \end{bmatrix}$$

Dependability Modeling with CTMCs

- Each state represents a particular error state, transition with component failure rate
  - States expresses number of failed components at any given time
  - Time-homogeneous process - Failure / repair rates do not change over time
  - Components have identical failure rates and identical repair rates
  - Failure and repair events are independent, process is memory-less
- Row sum is zero: Probability mass flowing out of state i will go to some other state
- Example:



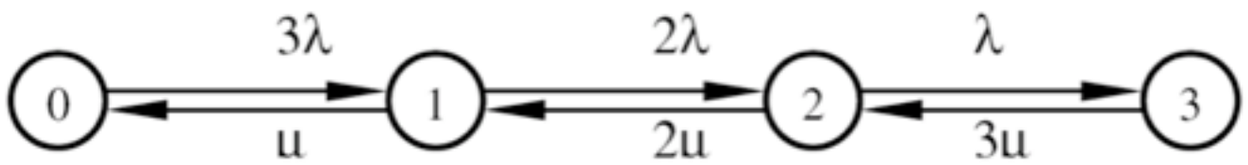
Example: 2-of-3 System



$$Q = \begin{pmatrix} -3\lambda & 3\lambda & 0 & 0 \\ \mu & -\mu - 2\lambda & 2\lambda & 0 \\ 0 & 2\mu & -2\mu - \lambda & \lambda \\ 0 & 0 & 3\mu & -3\mu \end{pmatrix}$$

- Interested in steady-state availability of the system
  - Interpretation as steady-state probability for the system being operational at t
  - Derived from *probability vector* -> contains steady-state probabilities for the system being in one of the failure states after a number of steps
- ‚Static‘ steady-state availability computable if probabilities are in equilibrium
  - Probability for leaving state is similar to probability for going into that state - probability mass is evenly distributed
  - Typically achieved after a high number of steps

Example: 2-of-3 System



$$3\lambda s_0 = \mu s_1$$
$$3\lambda s_0 + 2\mu s_2 = \mu s_1 + 2\lambda s_1$$
$$2\lambda s_1 + 3\mu s_3 = 2\mu s_2 + \lambda s_2$$
$$\lambda s_2 = 3\mu s_3$$
$$s_0 + s_1 + s_2 + s_3 = 1$$

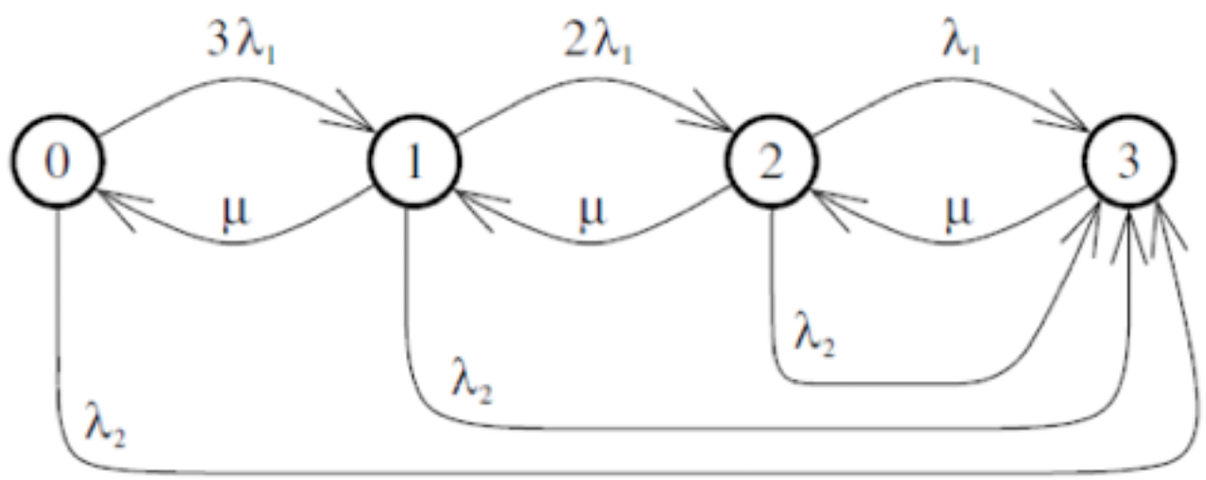
$$s_0 = \frac{\mu}{3\lambda} s_1$$
$$s_2 = \frac{\lambda}{\mu} s_1$$
$$s_3 = \frac{\lambda^2}{3\mu^2} s_1$$
$$s_1 = \frac{3\mu^2\lambda}{(\mu+\lambda)^3}$$

$$s_0 = \frac{\mu^3}{(\mu+\lambda)^3}; s_1 = \frac{3\mu^2\lambda}{(\mu+\lambda)^3}; s_2 = \frac{3\mu\lambda^2}{(\mu+\lambda)^3}; s_3 = \frac{\lambda^3}{(\mu+\lambda)^3}$$

$$A = s_0 + s_1 = \frac{\mu^2(\mu+3\lambda)}{(\mu+\lambda)^3} = 3a^2 + 2a^3 \qquad a = \frac{\mu}{(\mu+\lambda)}$$

Markov Chains

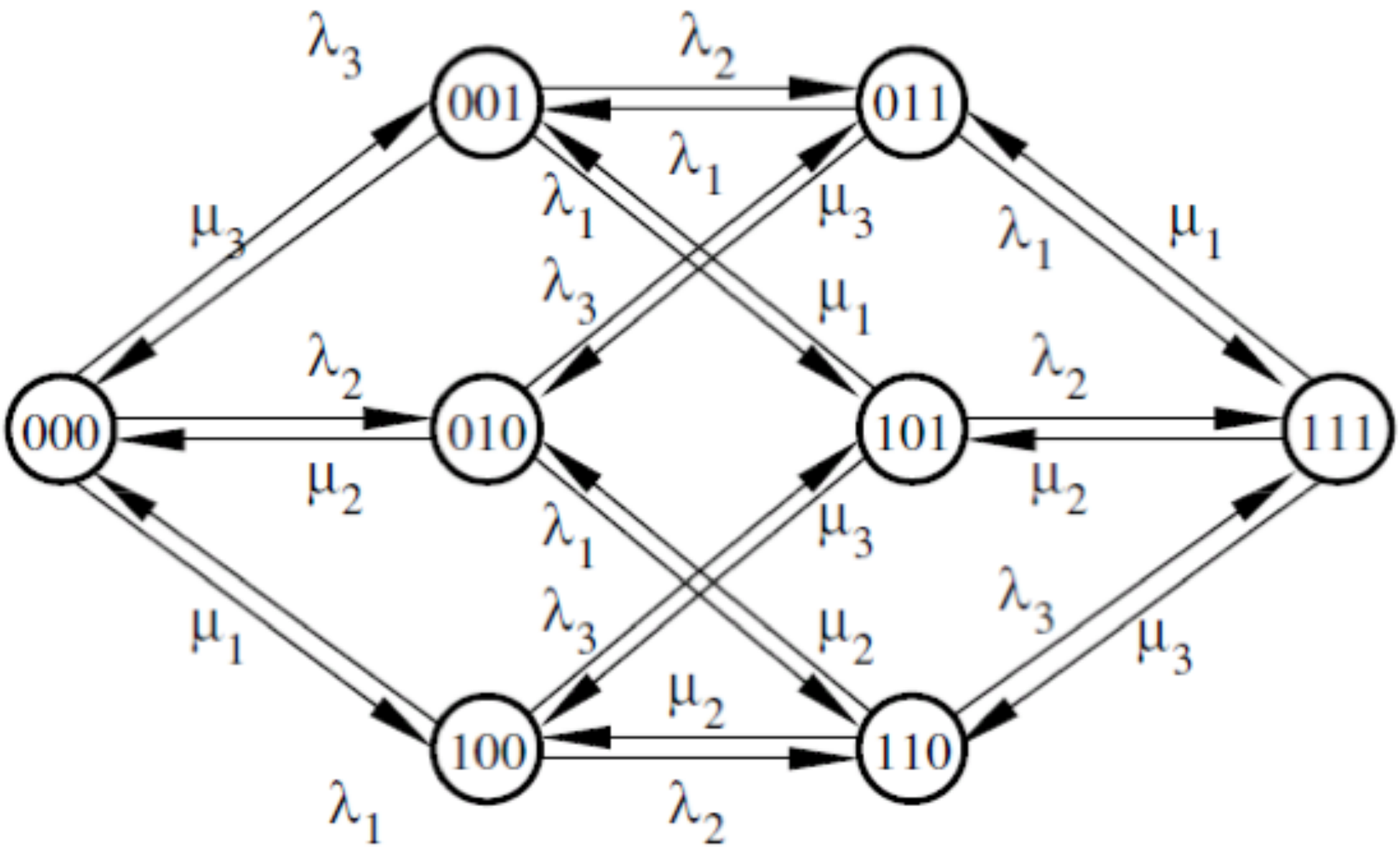
- Resulting formula equals to result from Boolean investigation, but Markov chains also support non-independent events - common cause failure



- Markov chains grow exponentially with their number of components - which is bad
  - *Divide-and-conquer* - Decompose and aggregate chain parts
  - *Structural decomposition* - Consider a system as set of independent subsystems
  - *Behavioral decomposition* - Assume time constants for some fault occurences and handling processes based on criticality - e.g. fault in parked airplane

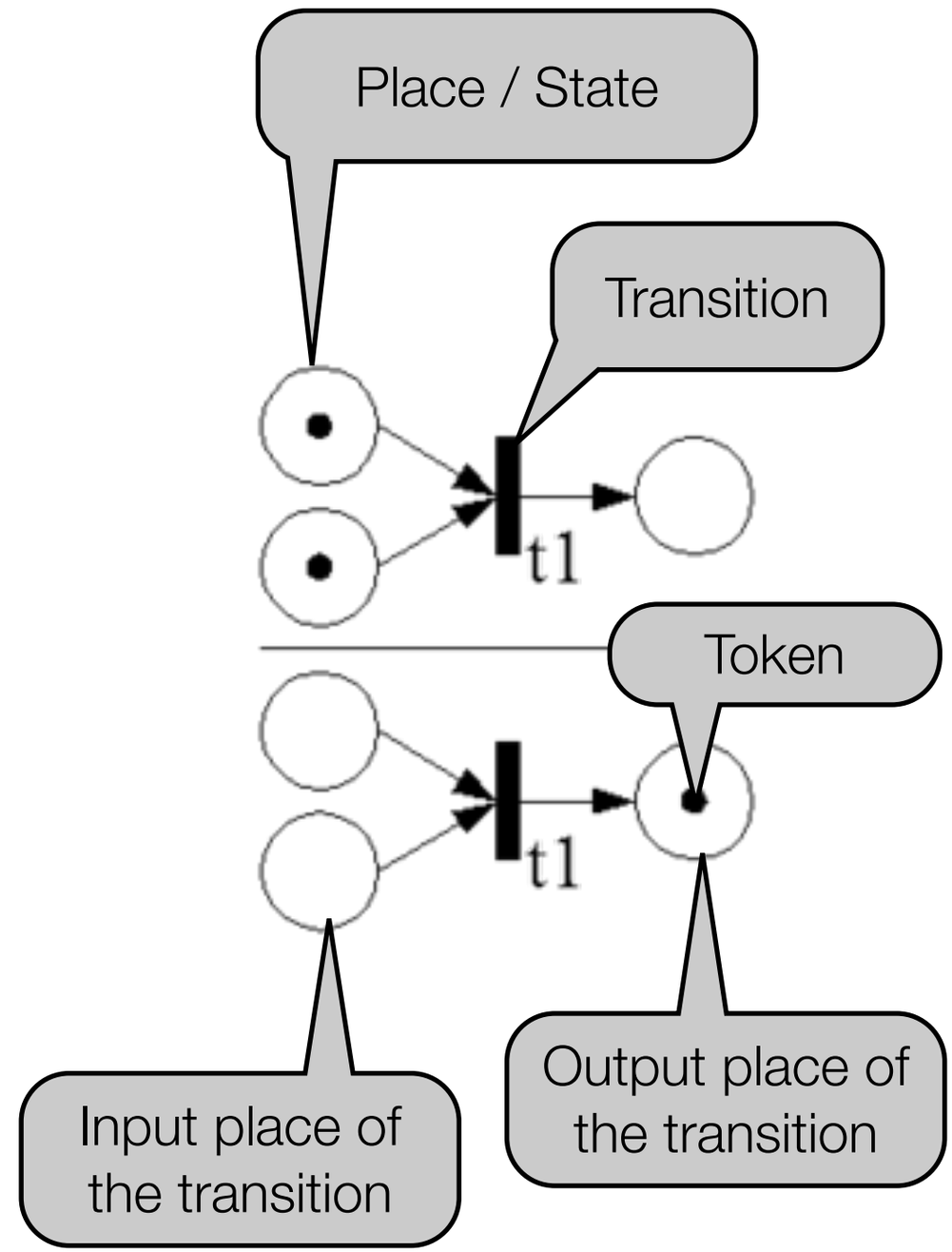
Example: Diverse Component Rates

- Models has 2^3 states, giving each a component a separate failure / repair rate



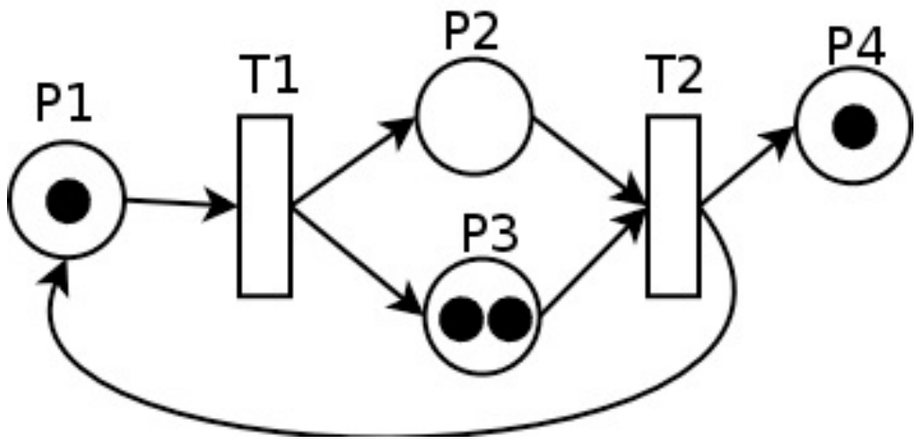
Stochatic Petri Nets

- Mathematical model for concurrent systems with many components (Carl Adam Petri)
- Bipartit directed graph (places vs. transitions)
- Each place has a capacity for tokens, default is unlimited or one
- Each arc has a weight expressing a cost factor, default is one
- Places are pre- / postconditions for transitions
- Distribution of tokens is called a **marking**
  - Every net has an **initial marking**



Stochastic Petri Nets

- Transition is **activated** (may fire) when
  - All input places contain enough tokens for the transition costs
  - All output places have enough capacity to take the new tokens
- Tokens are consumed and placed in output places, considering the arc weights
- Atomic nondeterminstic operation - any activated transition may fire
- Firing happens with given delay
- More complex Petri net versions can distinguish different token types
  - Colored tokens (data values)
  - Activation times for tokens

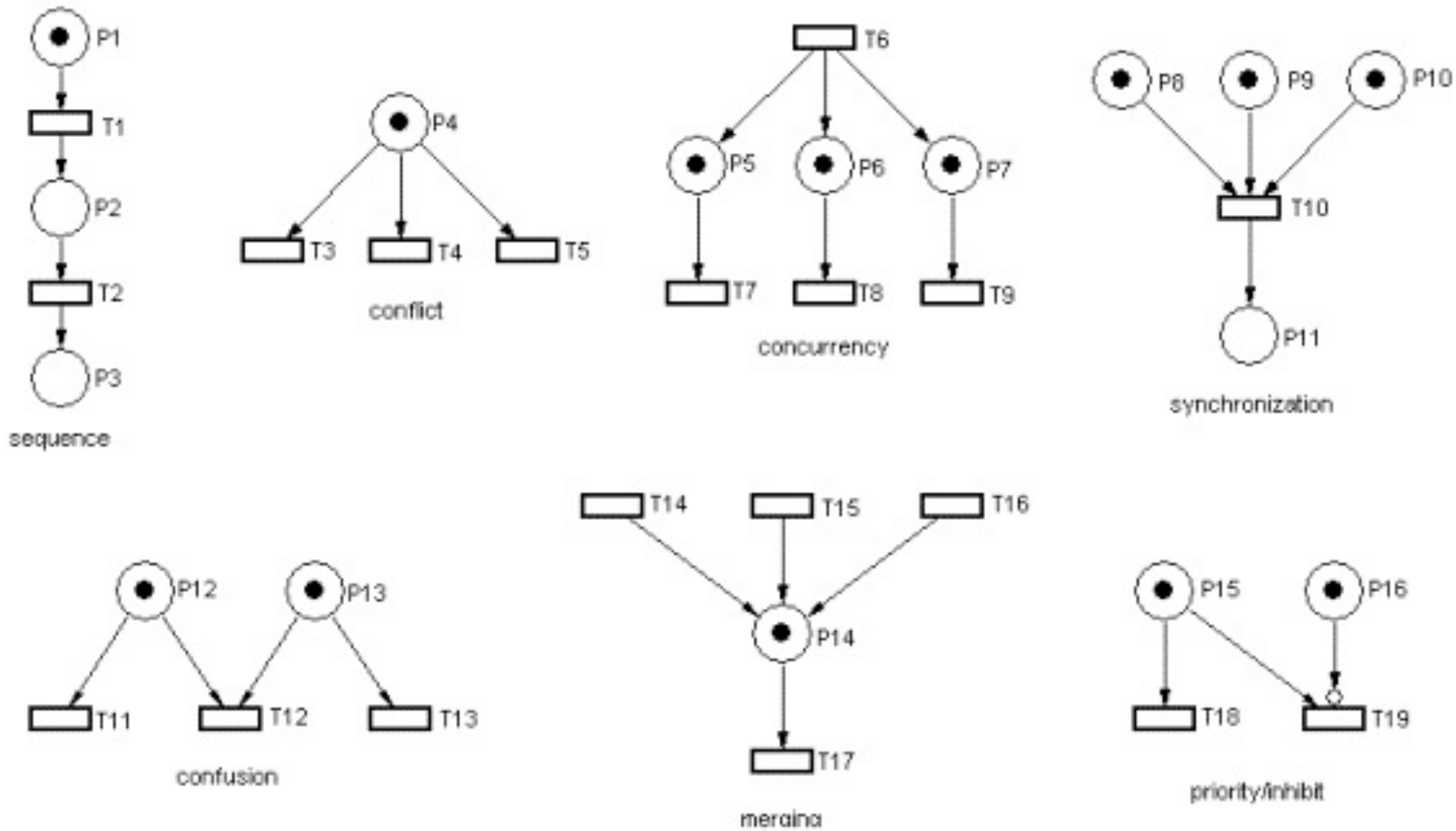


- Petri nets allow both formal analysis (for exponential distribution) and simulation

Petri Nets - Conceptional Mapping

Input Places	Transitions	Output Places
Required Resources	Task	Freed Resources
Input Data	Computations	Output Data
Input Signals	Signal Processing	Output Signals
Buffers / Registers	Processor	Buffers / Registers

Petri Net Concepts



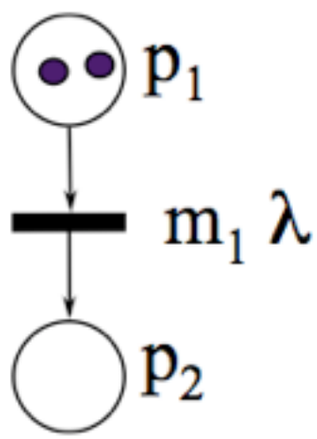
(C) Ming Chen

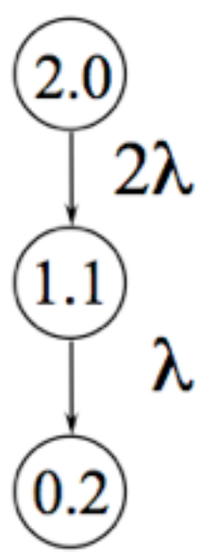
Typical Petri Net Properties

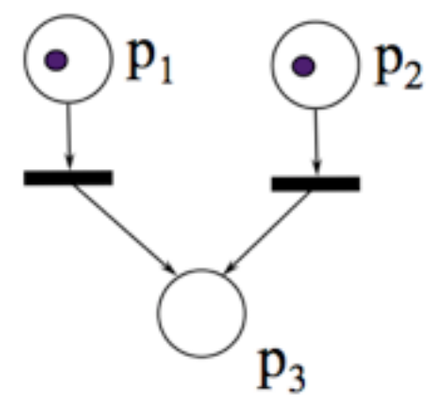
- **Reachability set**
  - All possible markings reachable from an initial marking
  - Possible analysis questions
    - Can some system state (e.g. an error state) be reached at all ?
    - Exists a firing sequence that transforms M_0 to M ?
- **Boundedness**
  - Marking is bounded if there is a k so that for every reachable marking the number of tokens in each place is bounded by k
  - Useful for modeling limited (bounded) resources

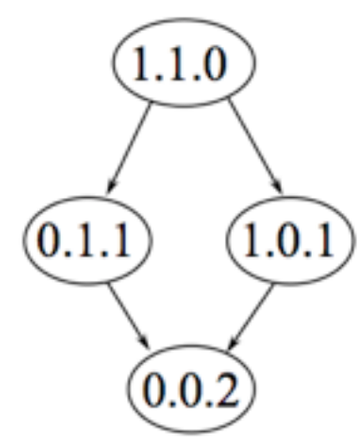
Petri Net -> Markov Chain

- Petri net has according reachability graph
  - Combines to Markov chain when transition probabilities are given

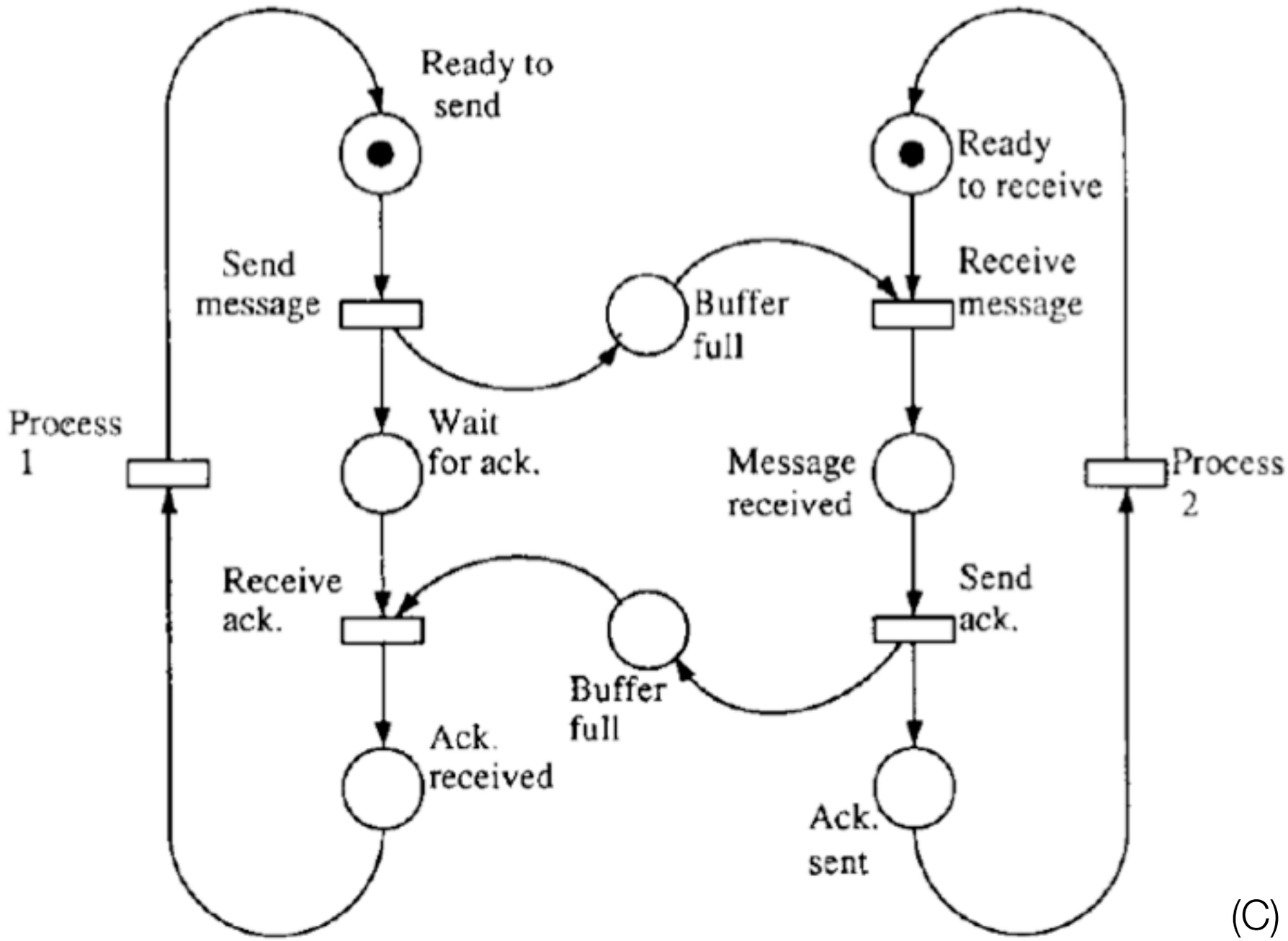






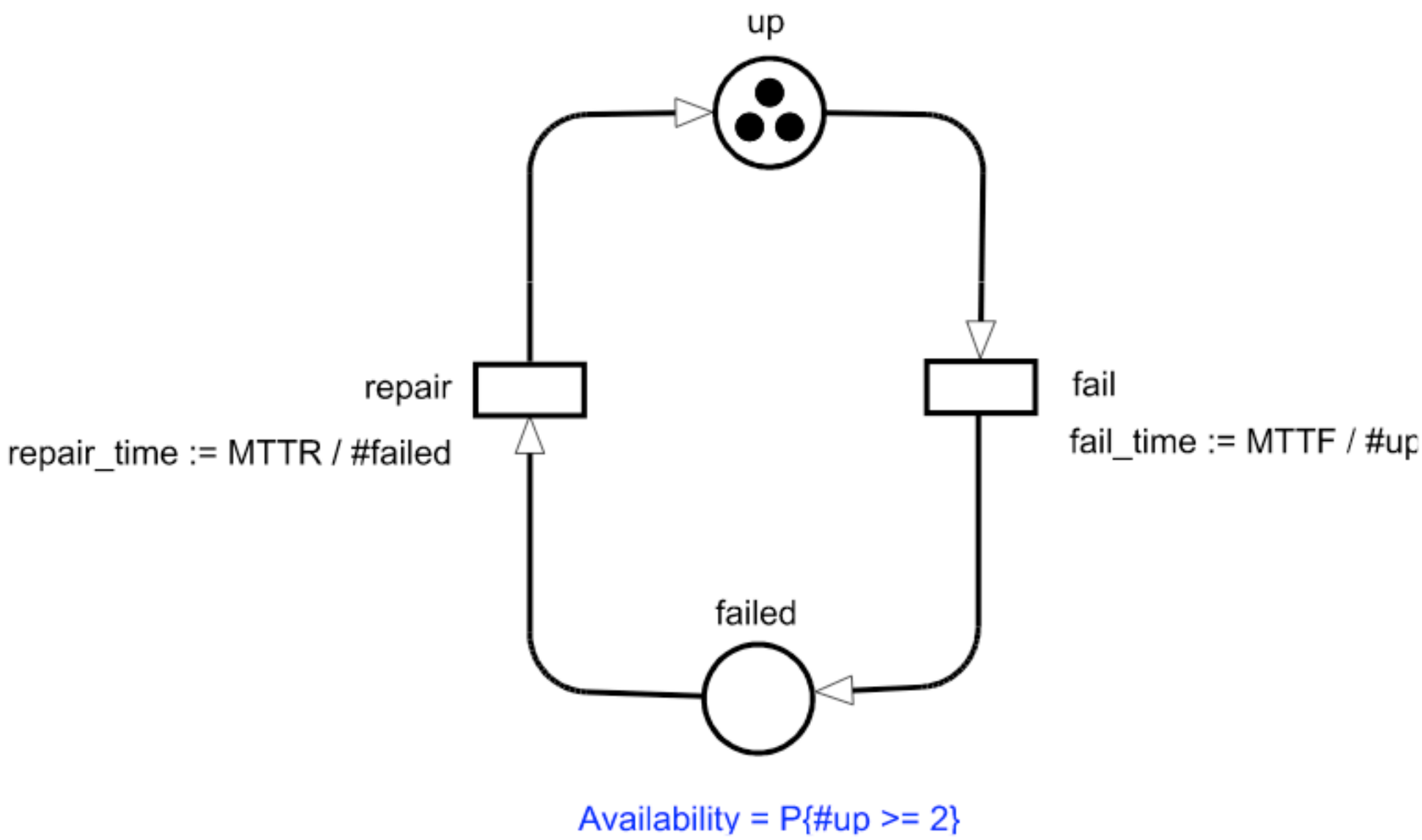


Example: Communication Protocol



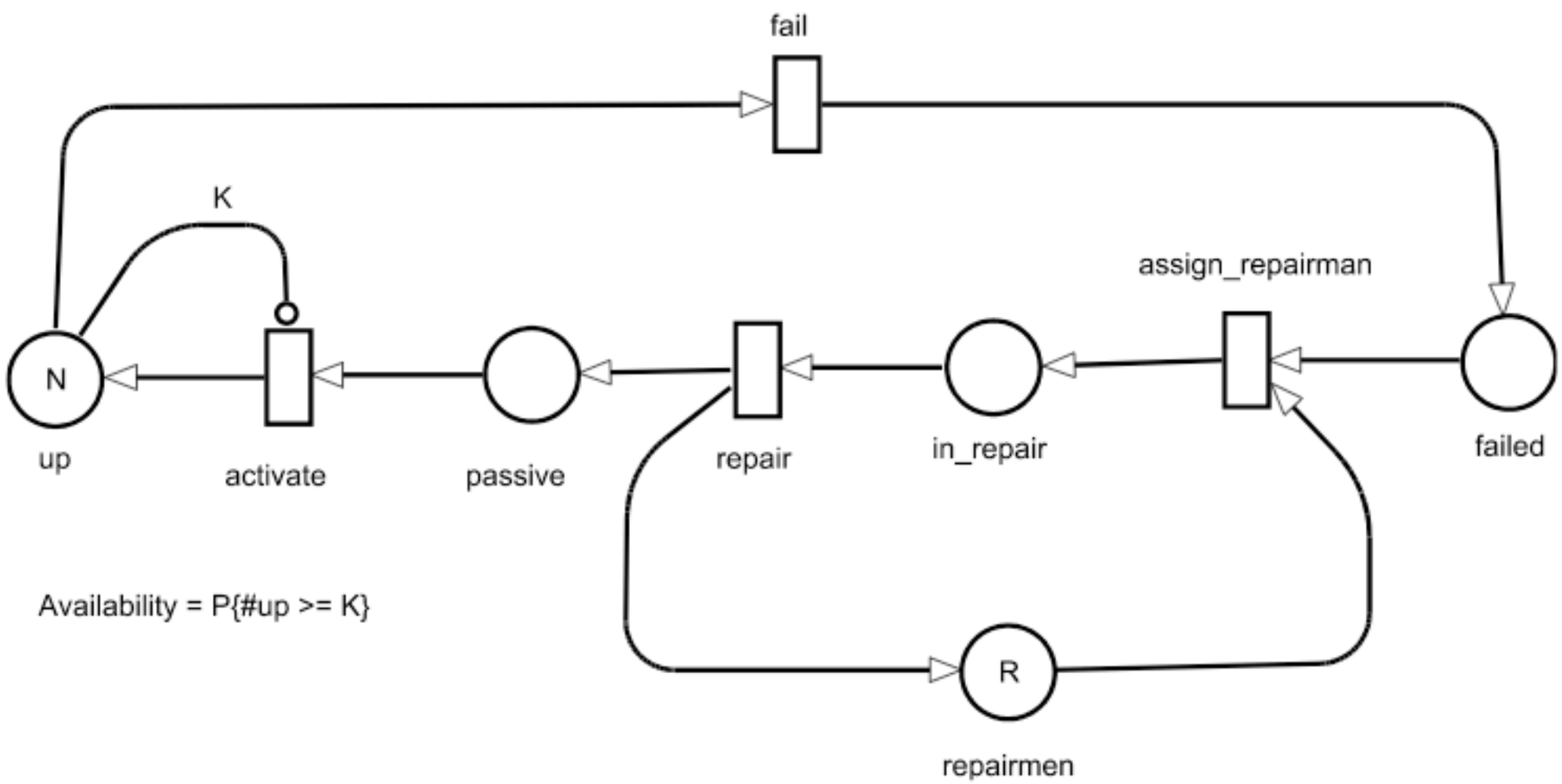
(C) A.W.Krings

Example: 2-of-3 System



Complexity of the petri net does not depend on the number of components !

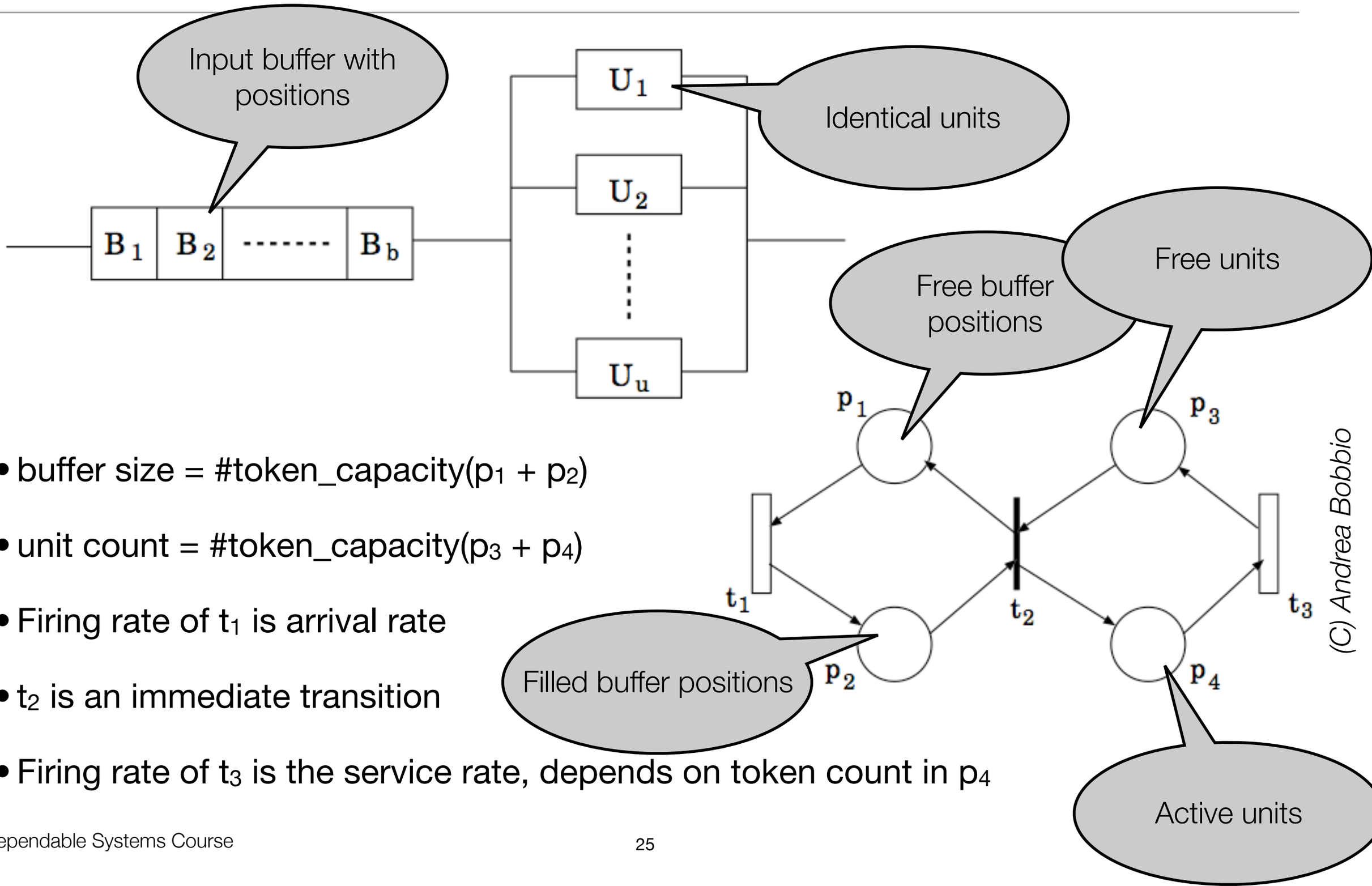
Example: K-of-N With Standby and Repairmen



Availability = P{#up >= K}

- Modeling of cold standby components (inhibitor arc)
- Limited repair capacities - at most R repairmen available at a time
- Dependability analysis - prove that there is no state where some property is violated

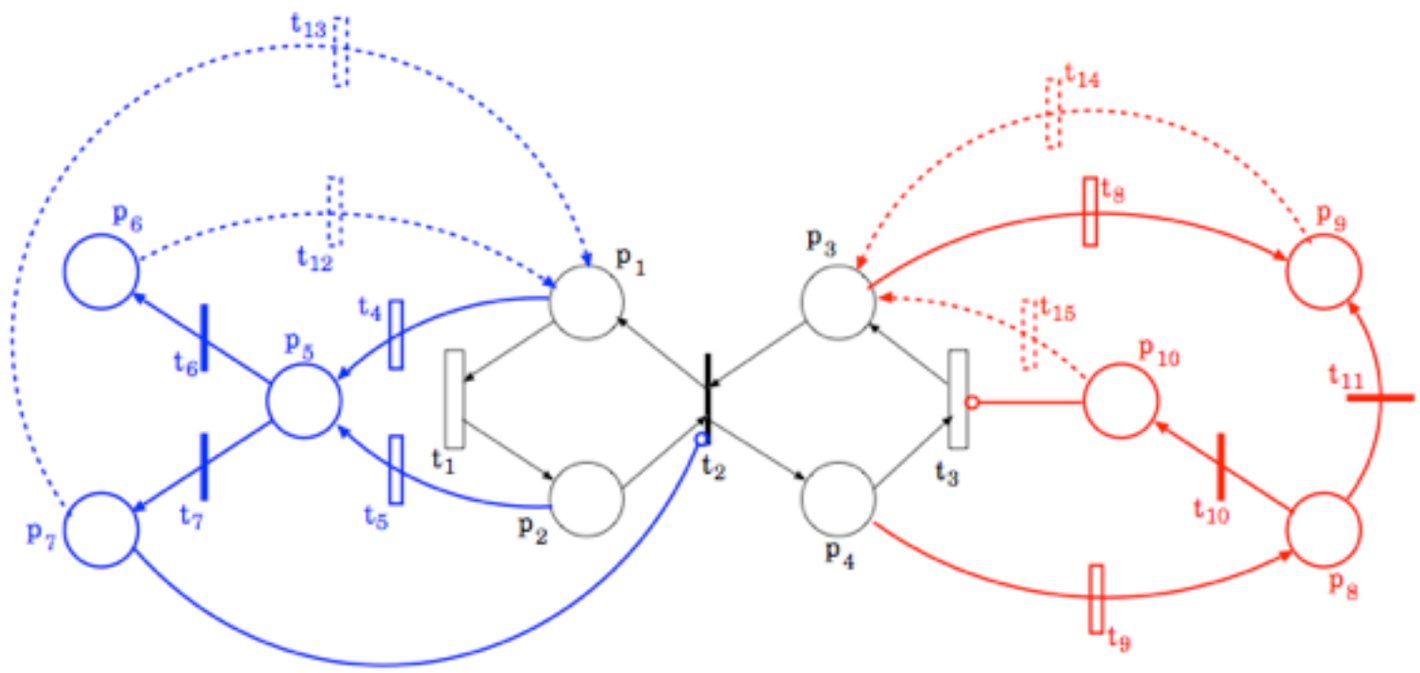
Example: Parallel System with Input Buffer



- buffer size = #token_capacity(p_1 + p_2)
- unit count = #token_capacity(p_3 + p_4)
- Firing rate of t_1 is arrival rate
- t_2 is an immediate transition
- Firing rate of t_3 is the service rate, depends on token count in p_4

Example: Parallel System with Input Buffer

p_1	Free buffer stage	
p_2	Occupied buffer stage	
p_3	Idle unit	
p_4	Active unit	
p_5	Failed buffer stage	
p_6	Recovered buffer stage failure	
p_7	Unrecovered buffer stage failure	
p_8	Failed active unit	
p_9	Recovered unit failure	
p_{10}	Unrecovered unit failure	
		firing rate
t_1	Buffer stage becomes occupied	λ
t_2	Transfer from buffer to unit	*immed.*
t_3	Unit ends a task	$m_4\,\mu$
t_4	Free buffer stage fails	$m_1\,\gamma_4$
t_5	Occupied buffer stage fails	$m_2\,\gamma_5$
t_6	Buffer stage failure is recovered	v_B
t_7	Buffer stage failure is not recovered	$(1 - v_B)$
t_8	Idle unit fails	$m_3\,\gamma_8$
t_9	Active unit fails	$m_4\,\gamma_9$
t_{10}	Unit failure is not recovered	$(1 - v_U)$
t_{11}	Unit failure is recovered	v_U
t_{12}	Repair of recovered buffer stage	ρ_{12}
t_{13}	Repair of unrecovered buffer stage	ρ_{13}
t_{14}	Repair of recovered unit	ρ_{14}
t_{15}	Repair of unrecovered unit	ρ_{15}



- Light lines - Fault free operation
- Heavy lines - Failures
- Dotted lines - repairs
- Rate computation demands exponential distribution

Petri Net Simulation

- In many cases, simulation is the only way to solve the net
  - More than one outgoing non-exponential distribution
  - Special guard functions
  - Complexity issues
  - ...
- Typical simulation problems
  - Modeled failure rates might be small, so many runs needed for valid result
  - Random number generation
  - Confidence intervals

Reliability Tools

System structure
Fault classes
Failure rates
Fault handling procedures
Repair procedures
Success criteria

Model development

Model solution (combinatorical, simulation)

Dependability evaluation

Fault Coverage
Reliability
Availability
Life Cycle
MTTF / MTTR
Costs
...

Errors in Dependability Evaluation (Salfner)