

EP2200 Queueing theory and teletraffic systems

Lecture 7

$M/M/m/m/C$ – Engset loss system

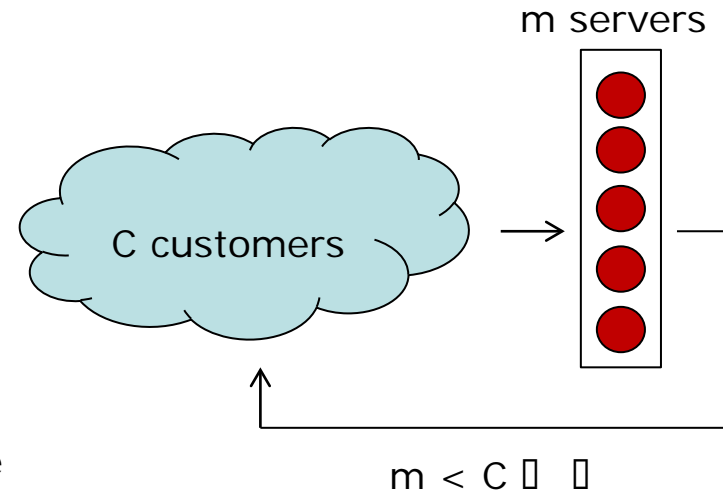
Viktorija Fodor
KTH EES/LCN

Markov queuing systems

- Systems considered until now:
 - M/M/1
 - M/M/1/K
 - M/M/m/m (Erlang loss system)
 - M/M/m (Erlang wait system)
- $\lambda_i = \lambda$
State independent Poisson arrival process:
to model independent requests from a large user population
- What happens if the population is small?
 - Finite population system:
 - M/M/m/m/C – Engset loss system
 - General case with buffer on the recitation

M/M/m/m/C

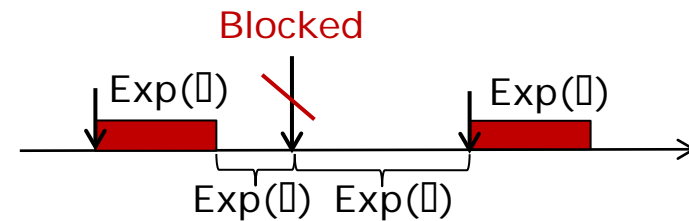
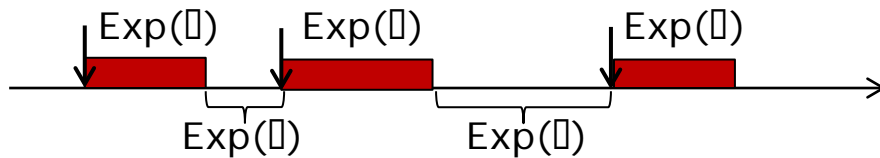
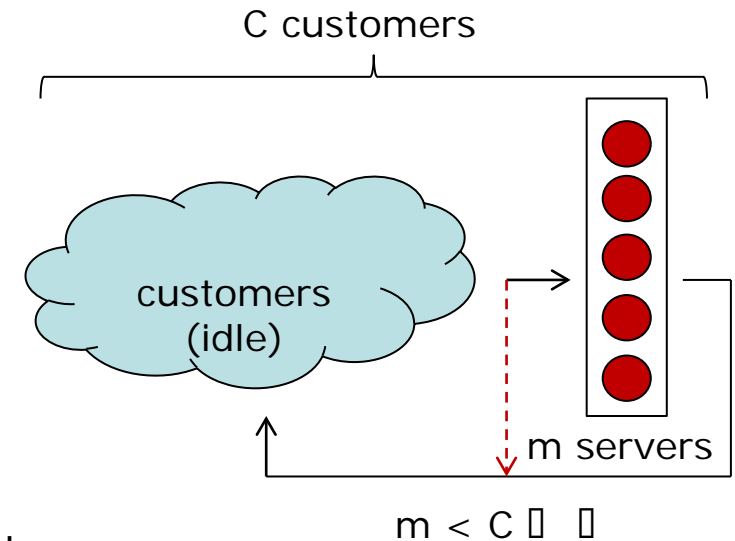
- System definition
 - C customers
 - m servers
 - no buffer
 - Exponential service time (μ)
 - Arrival process should reflect the finite population: customer does not generate new request when one is under service □
the intensity of new request arrivals depends on the number of requests under service



(Example: you do not start new phone calls when you already talk to someone)

M/M/m/m/C

- Modeling finite population
- Single customer behavior:
 - Thinking (idle) time after served request: $\text{Exp}(\lambda)$ (Virtamo notes: λ)
 - Request service time: $\text{Exp}(\mu)$
 - Blocked request: does not try again, moves to idle state



M/M/m/m/C

- State transition diagram (Markov chain)
- State probabilities in steady state:
 - Engset distribution:

$$p_k = \frac{\binom{C}{k} \left(\frac{\lambda}{\mu}\right)^k}{\sum_{i=0}^m \binom{C}{i} \left(\frac{\lambda}{\mu}\right)^i}$$

- Probability that the arriving node finds the system in state k :
PASTA does not hold!

$$a_k = \frac{\lambda_k p_k}{\sum_{i=0}^m \lambda_i p_i}$$

- Time blocking = part of the time the system is in blocking state = p_m
- Call blocking = P(arriving request gets blocked) = a_m

M/M/m/m/C

- Performance

- Offered traffic:

$$\lambda^* = \sum_{i=0}^m (C-i)\lambda p_i$$

- Effective traffic:

$$\lambda_{eff} = \sum_{i=0}^{m-1} (C-i)\lambda p_i$$

- Average number of requests under service:

$$N = N_s = \lambda_{eff} / \mu$$

- When should we consider a system as finite population system?
 - Rule of thumb: $C > 10m$

Summary

- Markovian systems:
 - Poisson arrival (may be state dependent)
 - Exponential service time (may be state dependent)
 - One or many servers
 - No buffer, limited or infinite buffer space
- Next step
 - Non-exponential service
 - Systems still possible to model with a Markov chain
 - Erlang and Hyper-Exponential service times