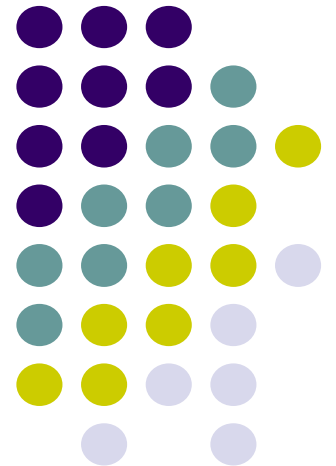


A Model for Analysis of SYN Flood DoS Attacks

Nimal Nissanke and Jun Sun

London South Bank University, London



Aims and Objectives



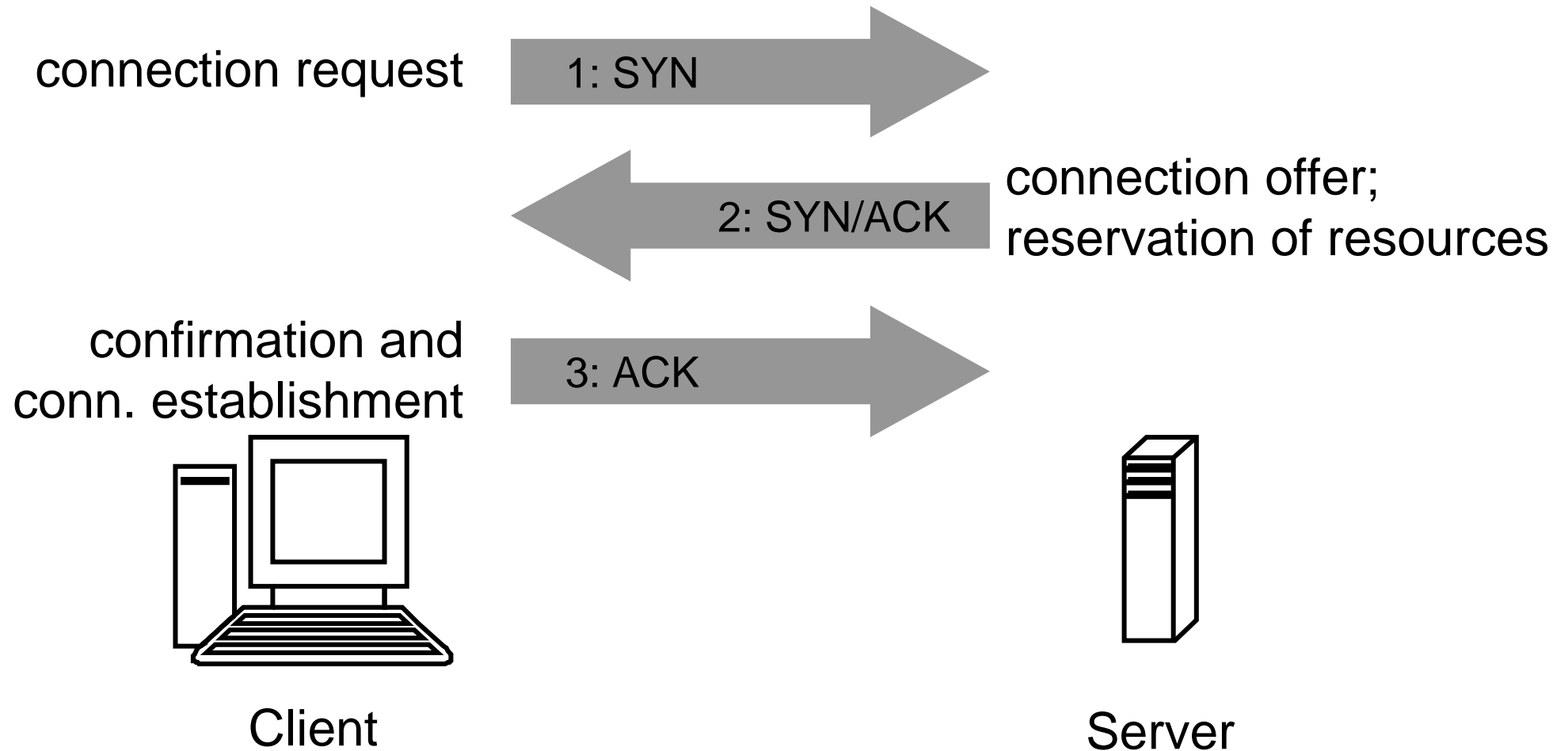
- To formulate a framework for **analysis of SYN flood DoS attacks**
- To formulate a **new approach**
 - Based on the underlying mathematical model which is capable of:
 - Early and accurate detection of DoS attack
 - Improving server side defences
 - Devising ways to minimise disruptions to legitimate users
 - Establishing **worst-case scenarios** of attack traffic and background network traffic for better defences

An Overview

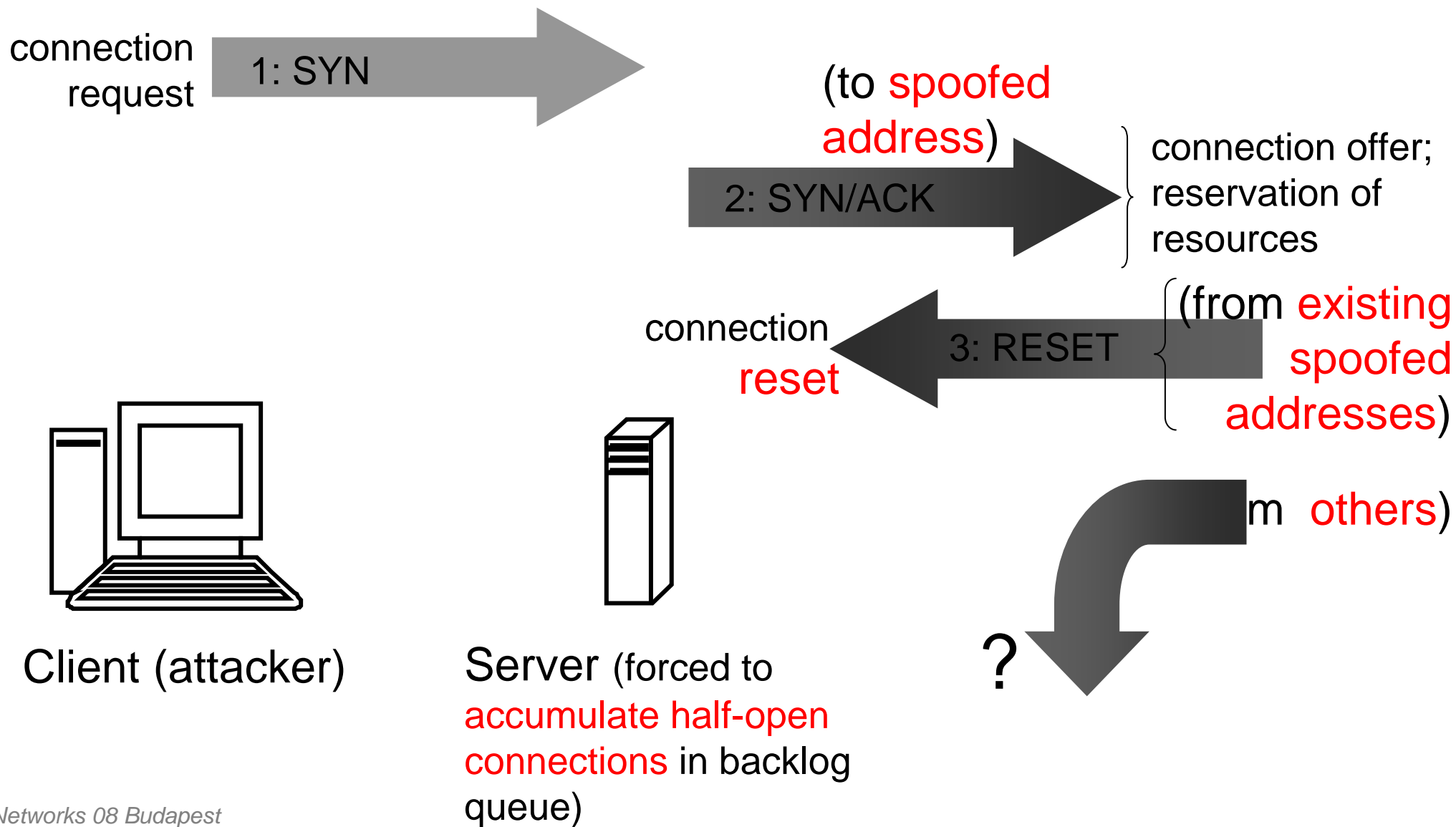
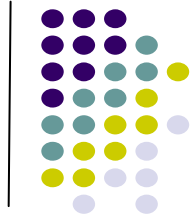


- A brief note on SYN flood DoS attacks
- An **analytical framework (the focus)**
 - a **simple case**: a closed form solution for **Poisson arrivals** of connection requests
 - the **general case: simulation** for any form of arrival of connection requests
 - A few numerical results (to show the potential)
- (background, overall context and related works – see the paper)
- Conclusions

TCP's 3-way Handshake Connection Establishment Protocol



SYN-flood attack



Requirements of an Effective Mechanism



- A **high level of system security** (not adversely affecting the delivery of legitimate services)
- Capability to provide **early** and **accurate detection**
- Capability to **reduce the intensity** of any suspected attack traffic significantly and swiftly upon detection
- Capability to **avert** or **disable attacks**
- **Small** or no **changes** to the current existing **network infrastructure**
- **Minimal degradation** of network **performance** when attack detection and defense measures are deployed

About the Mathematical Model



Its aims:

- Produce a model of the **server side TCP** for dynamically adjusting **tunable parameters** (to speed up discarding of half-open attack connections)
- Underlying the model is an **attack detection algorithm** based on **distribution of request arrivals**, **RTT** (round trip time) **distribution** and **connection failure patterns**
- The purpose of the **mathematical model**
 - To develop the **heuristic rules** for the **deployment** of the technique in **detecting and combating attacks**

About the Mathematical Model (Strategy)



- Two environmental factors
 - The rate and nature of incoming connection requests (SYN traffic)
 - Their RTTs (round trip times) (high/low)
- A necessary condition: high-intensity incoming SYN traffic
- A sufficient indication of a (high-intensity) attack:
 - The above + High occupancy of backlog queue with high RTT connections
 - Because server could work properly even under high-intensity traffic with relatively low RTT values

Backlog Queue Model: Behaviour and Parameters

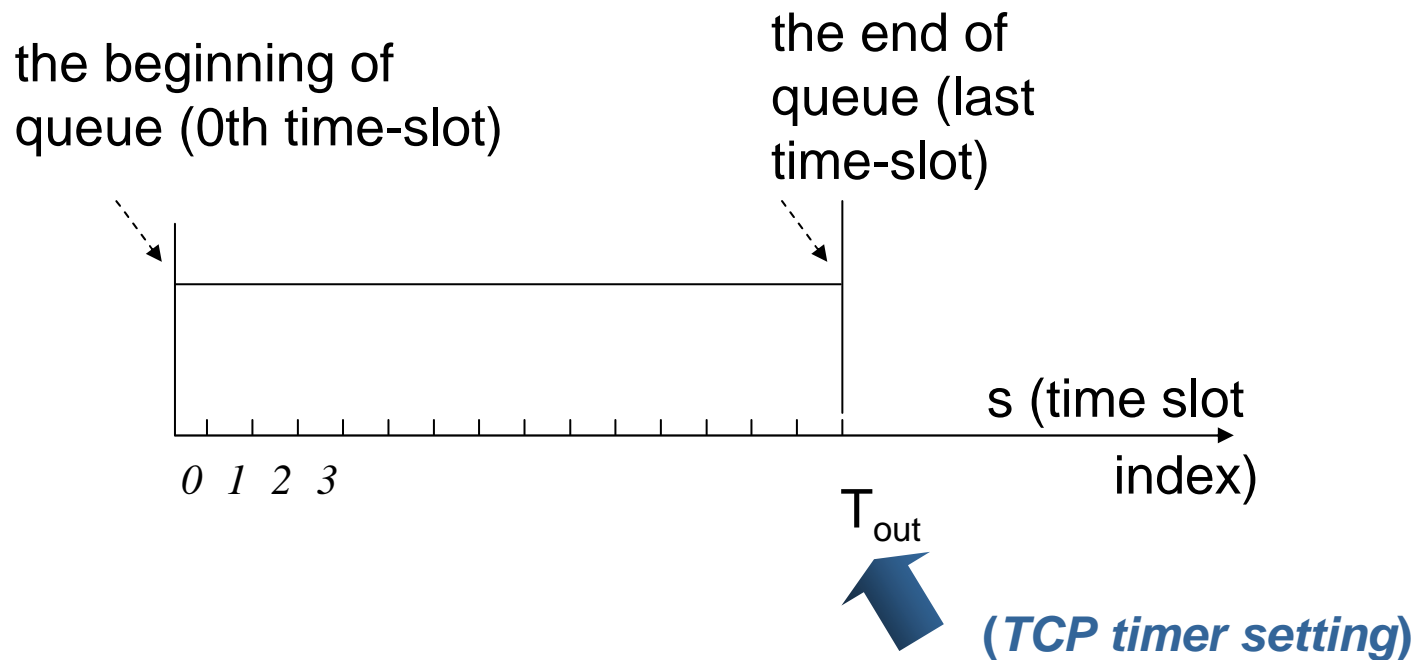


- Behavioural Characteristics
 - occupancy level
 - Time-out failures (connection requests that are timed out)
 - Admission (buffer) failures (connection requests failing to gain admission because of overcrowding)
- Parameters
 - Environmental parameters
 - a) A statistical traffic model of incoming connection requests
 - b) The distribution of Round Trip Time (RTT)
 - System Parameters
 - c) Size of the backlog queue (B_{max})
 - d) TCP's time-out parameter (T_{out})

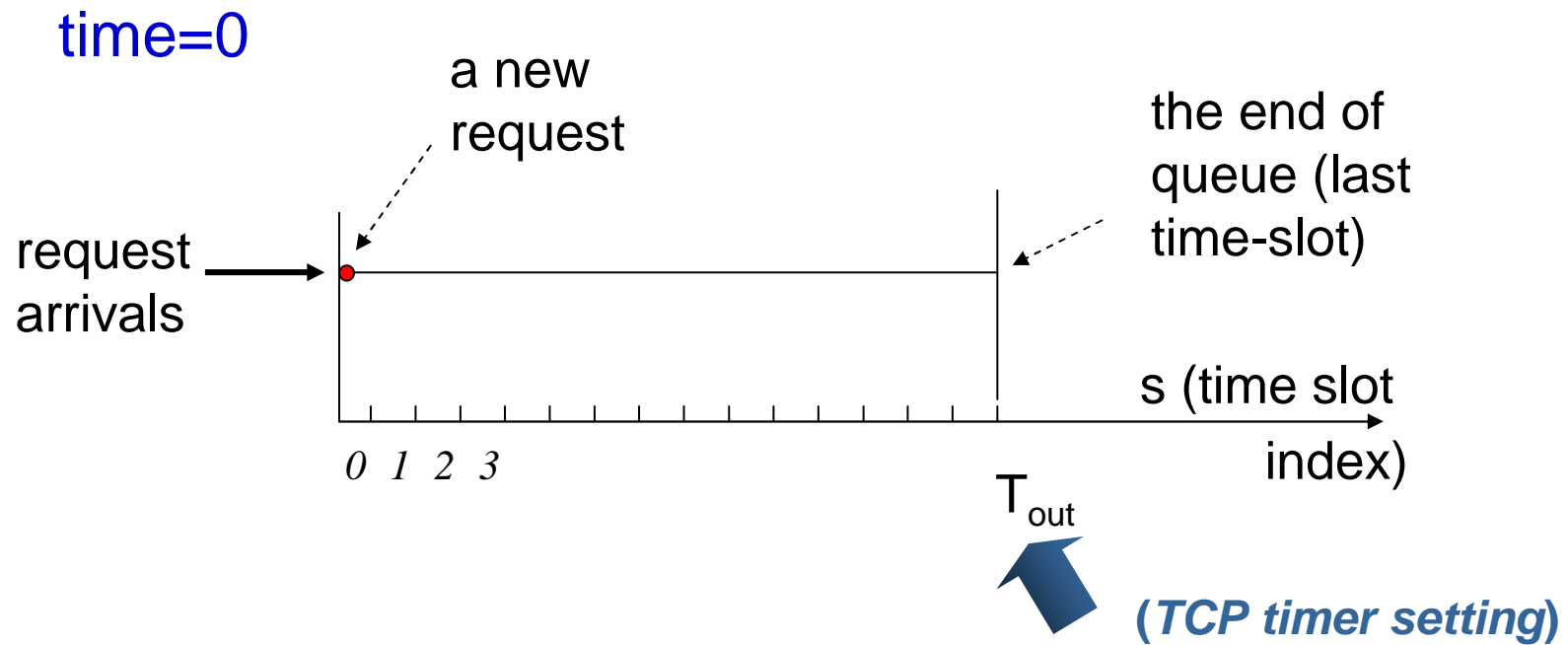
The Model ... Visually



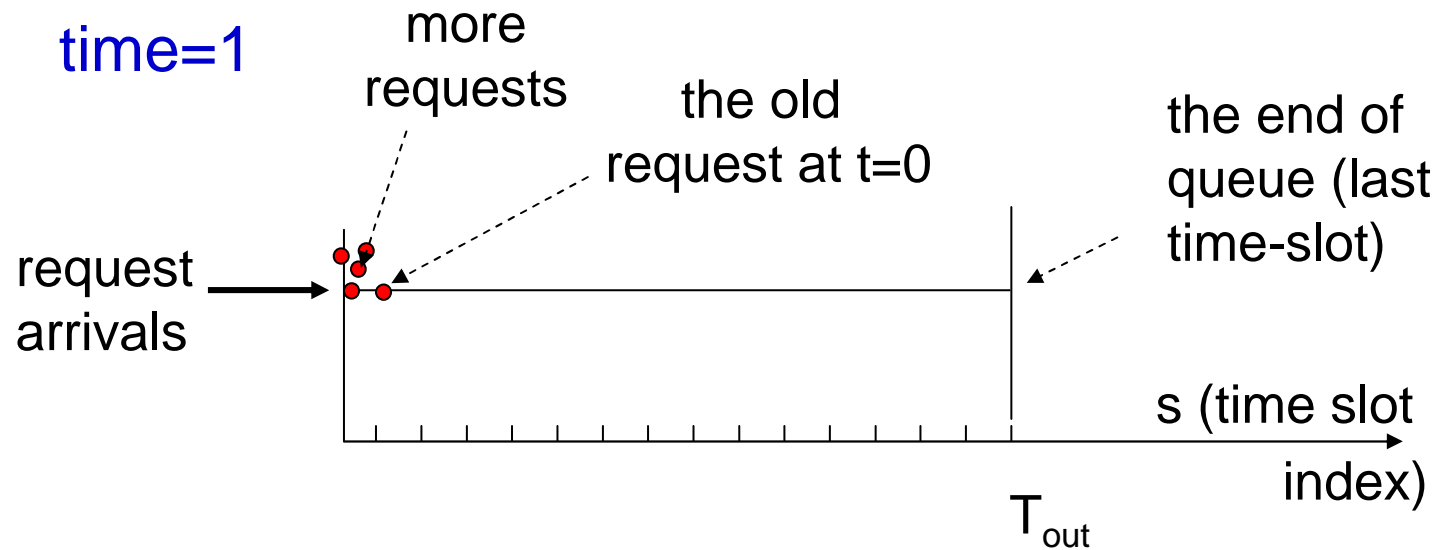
A 'Time-slotted' Backlog Queue



The Model ... Visually



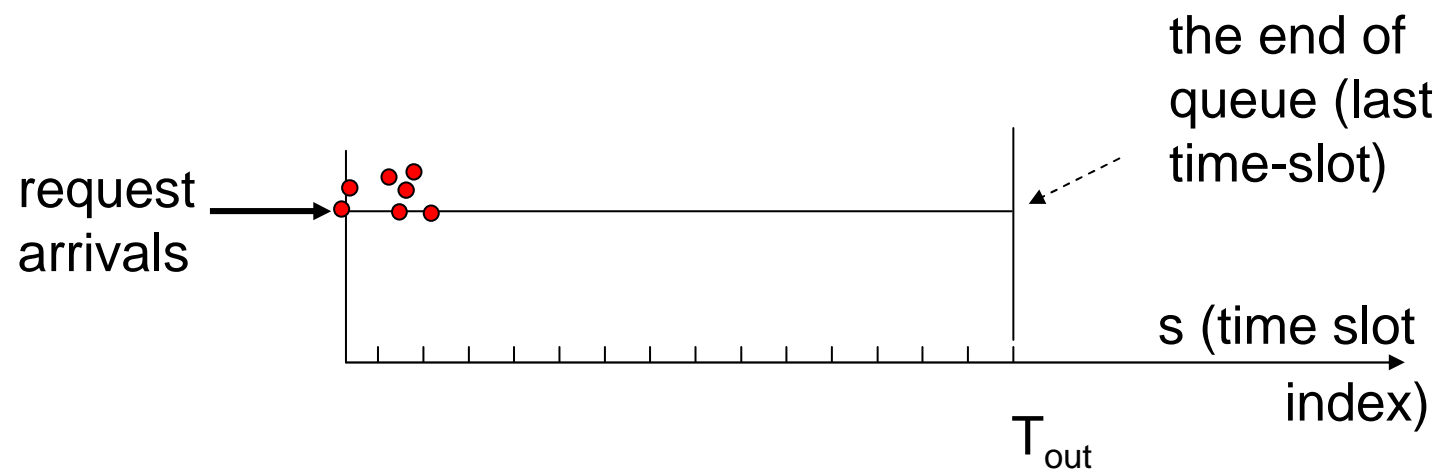
The Model ... Visually



The Model ... Visually



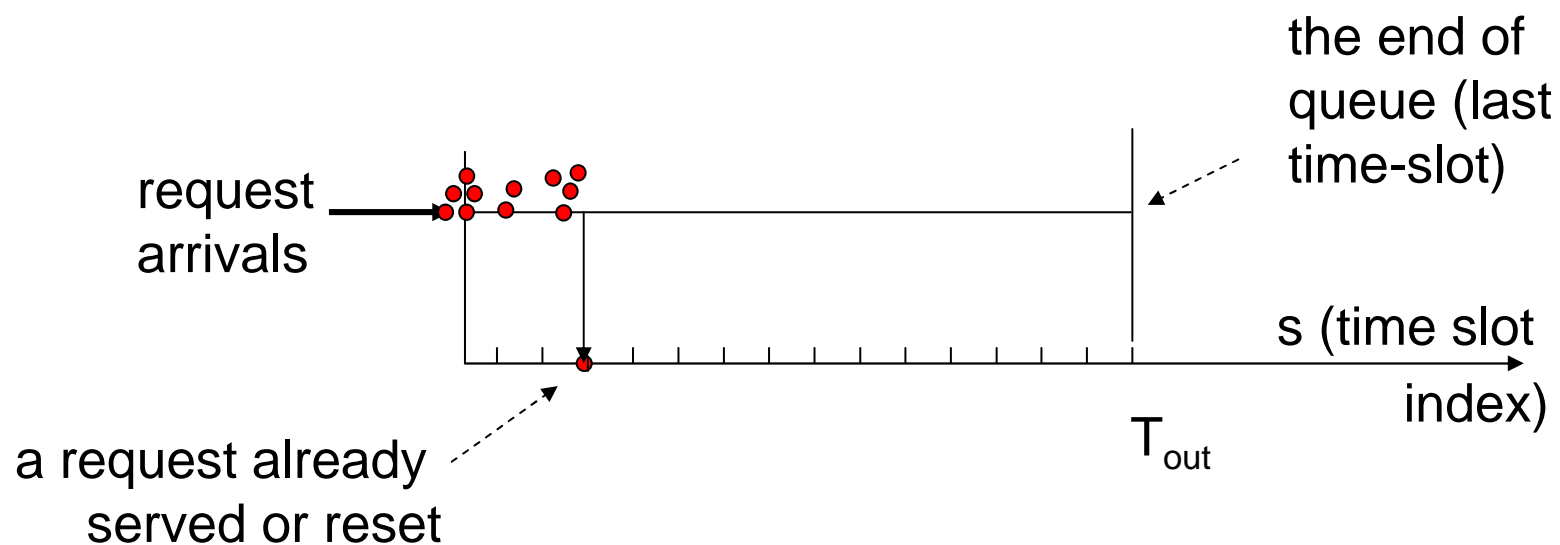
time=2



The Model ... Visually

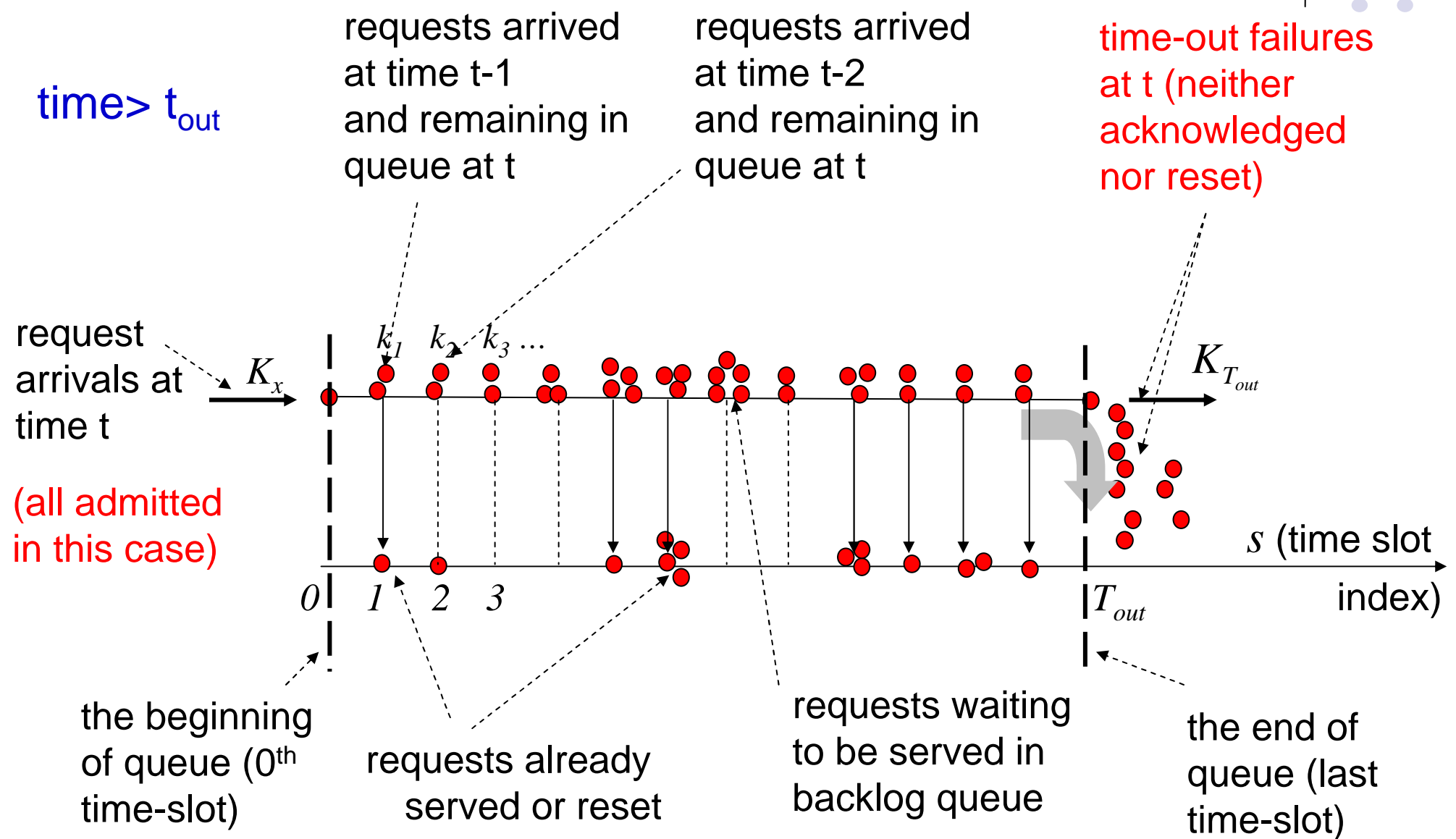


time=3



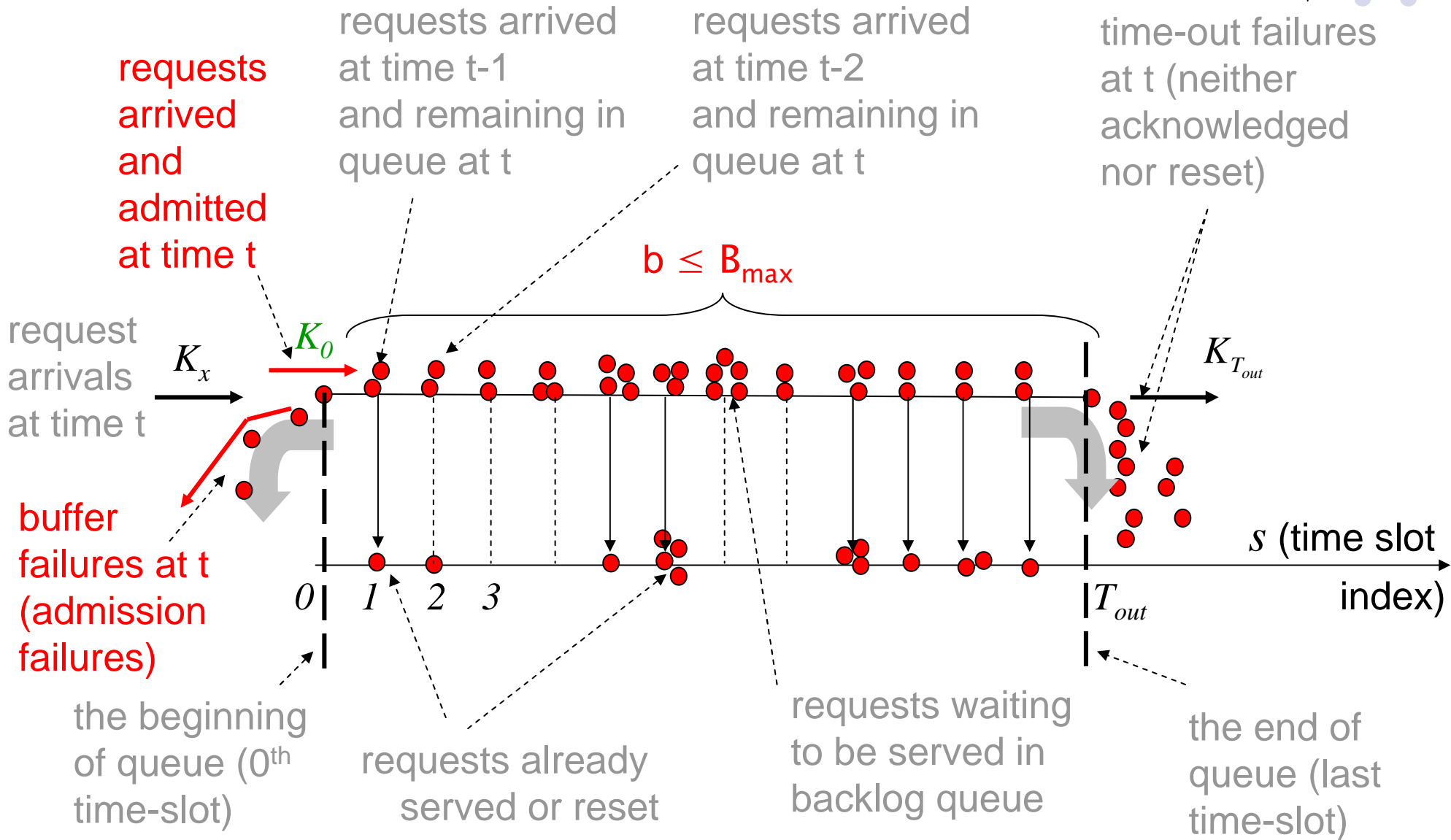


The Model ... Visually (with *Infinite Buffer*)

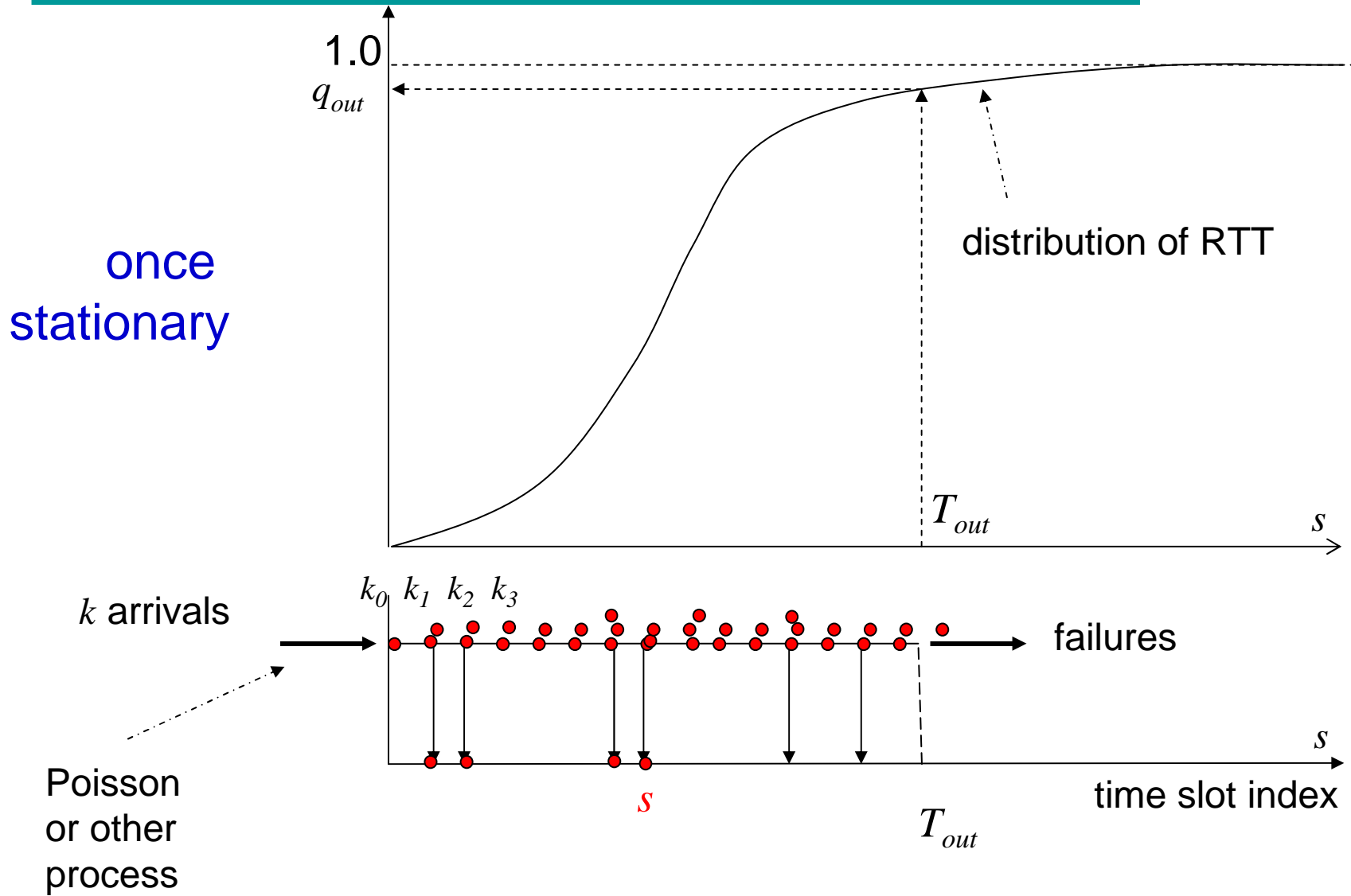




The Model ... Visually (with *Finite Buffer*)

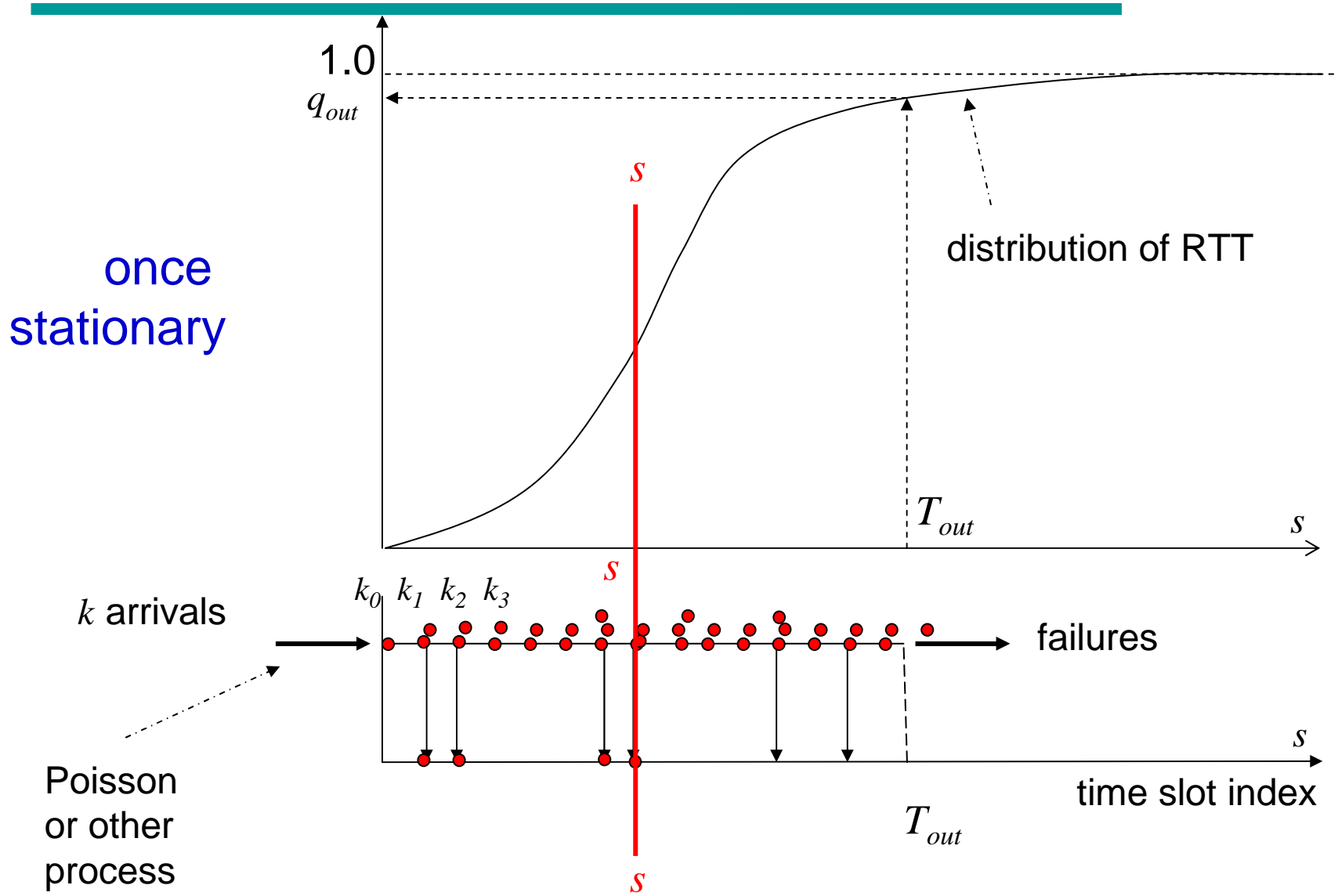


The Mathematical Notation



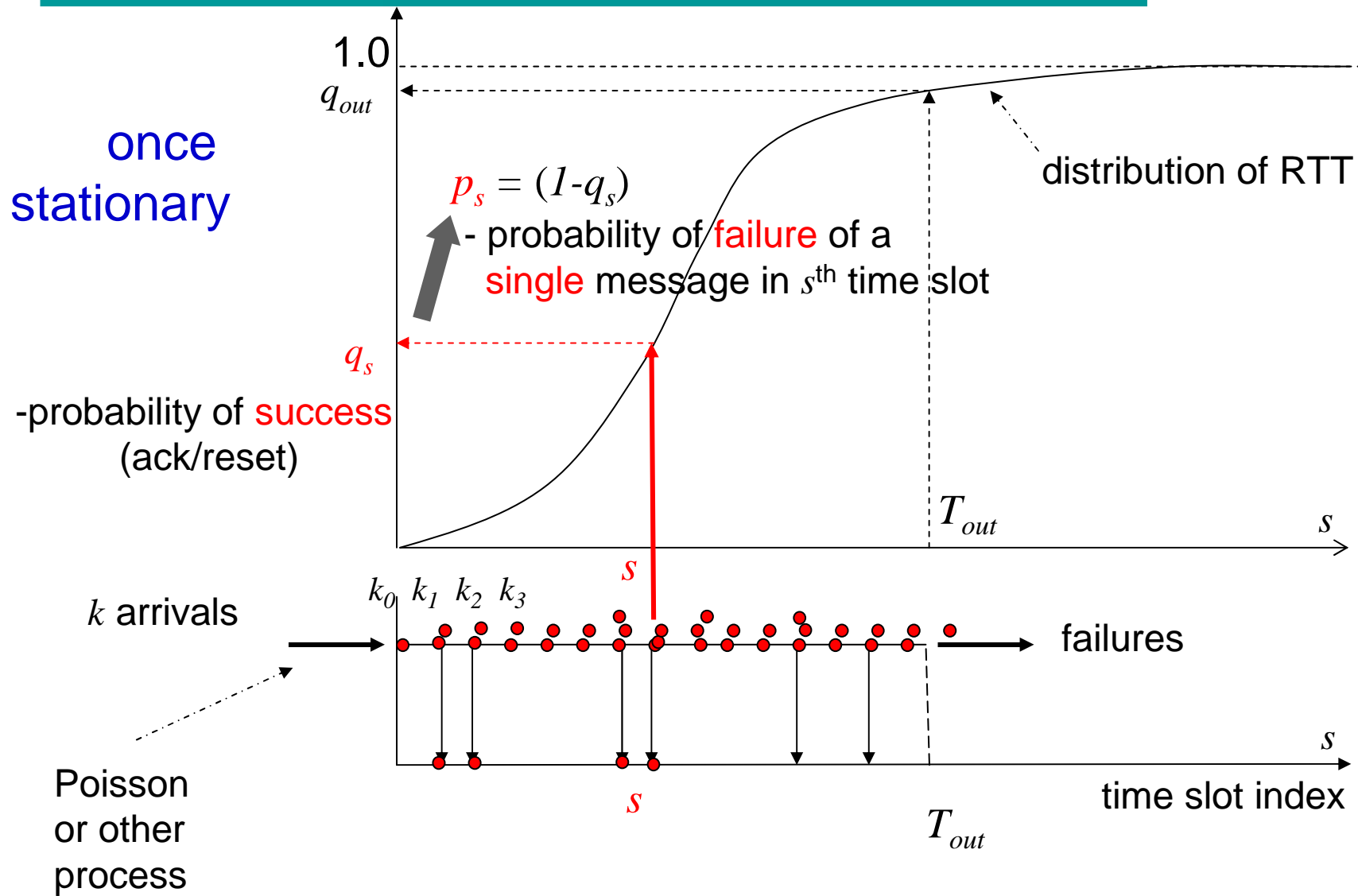


The Mathematical Notation



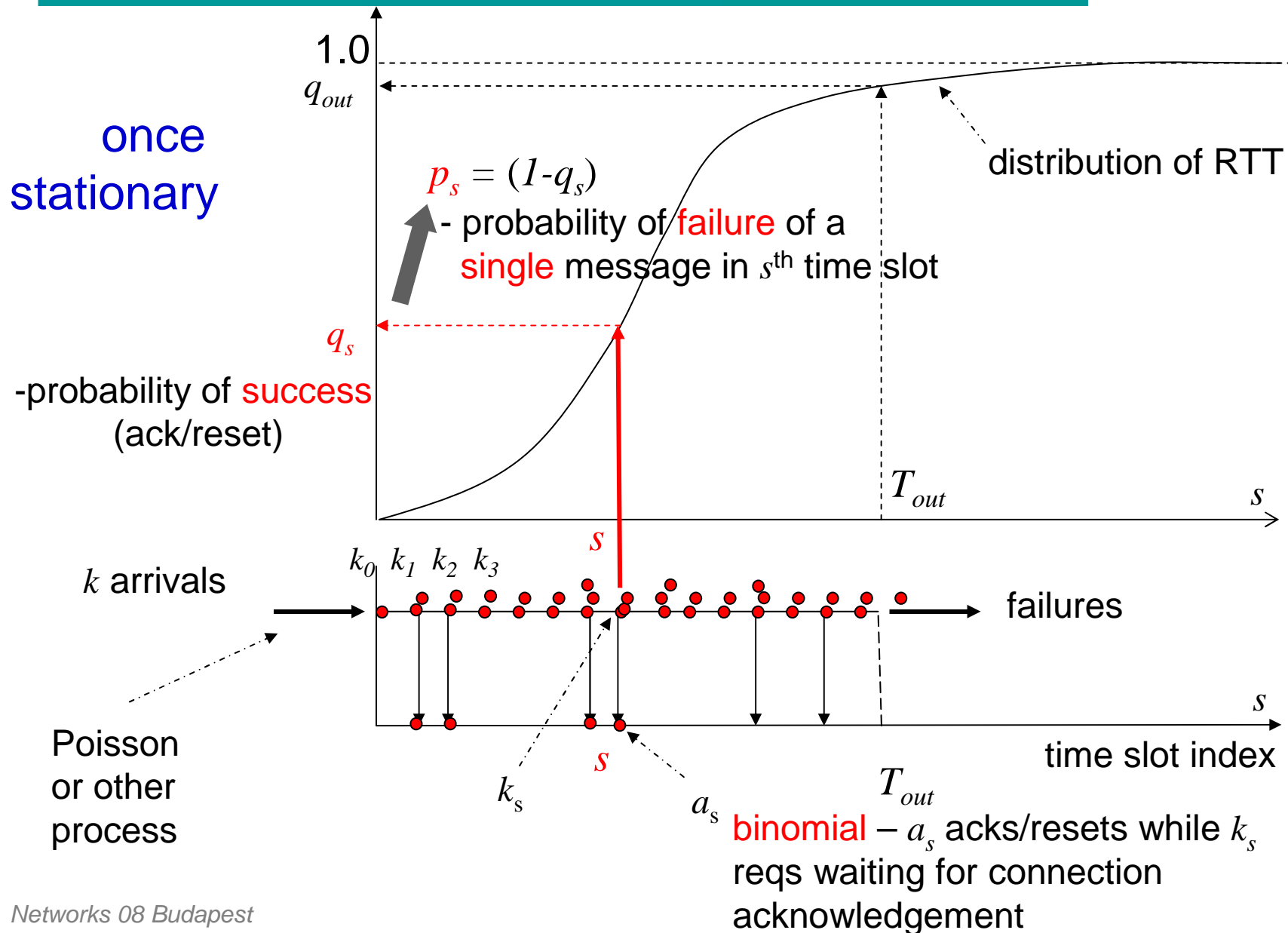


The Mathematical Notation





The Mathematical Notation



Mathematically (with *Infinite Buffer*)



- K_x – the number of incoming SYN-packets (requests) - a random variable) made over a time interval x
- Consider a Poisson process for requests at a rate λ over the length time x

$$P(K_x = k) = P_{K_x}(k) = \frac{(\lambda x)^k}{k!} e^{-\lambda x} = \frac{\lambda^k}{k!} e^{-\lambda}$$

letting x to be a unit time ($x=1$)

- Model for acknowledgements and resets - Binomial distribution (k_s Bernoulli trials: the probability of i successes out of k_s requests in s -th time slot, given that each request has a probability p_s of success)

$$\binom{k_s}{i} p_s^i (1 - p_s)^{k_s - i}$$



Mathematically (with *Infinite* Buffer)

Probability of A_s and K_s taking values i and j :

$$\begin{aligned} P(A_s = i, K_s = j) &= P(A_s = i, K_s = j \mid K_{s-1} = i + j) \\ &= \binom{i+j}{i} p_s^i (1-p_s)^j \frac{\lambda_{s-1}^{i+j}}{(i+j)!} e^{-\lambda_{s-1}} = P_{A_s} P_{K_s} \end{aligned}$$

where

$$P_{A_s}(i) = P(A_s = i) = \frac{(p_s \lambda_{s-1})^i}{i!} e^{-p_s \lambda_{s-1}}$$

$$P_{K_s}(k) = P(K_s = j) = \frac{(q_s \lambda_{s-1})^j}{j!} e^{-q_s \lambda_{s-1}} = \frac{(\bar{q}_s \lambda_0)^j}{j!} e^{-\bar{q}_s \lambda_0}$$

$$q_s = (1 - p_s) \text{ and } \bar{q}_s = \prod_{m=0}^s q_m$$

Mathematically (with *Infinite Buffer*)



Probability of A_s and K_s taking values i and j :

$$P(A_s = i, K_s = j) = P(A_s = i, K_s = j | K_{s-1} = i + j) \\ = \binom{i+j}{i} p_s^i (1-p_s)^j \frac{\lambda_{s-1}^{i+j}}{(i+j)!} e^{-\lambda_{s-1}} = P_{A_s} P_{K_s}$$

where

$$P_{A_s}(i) = P(A_s = i) = \frac{(p_s \lambda_{s-1})^i}{i!} e^{-p_s \lambda_{s-1}}$$

$$P_{K_s}(k) = P(K_s = j) = \frac{(q_s \lambda_{s-1})^j}{j!} e^{-q_s \lambda_{s-1}} = \frac{(\bar{q}_s \lambda_0)^j}{j!} e^{-\bar{q}_s \lambda_0}$$

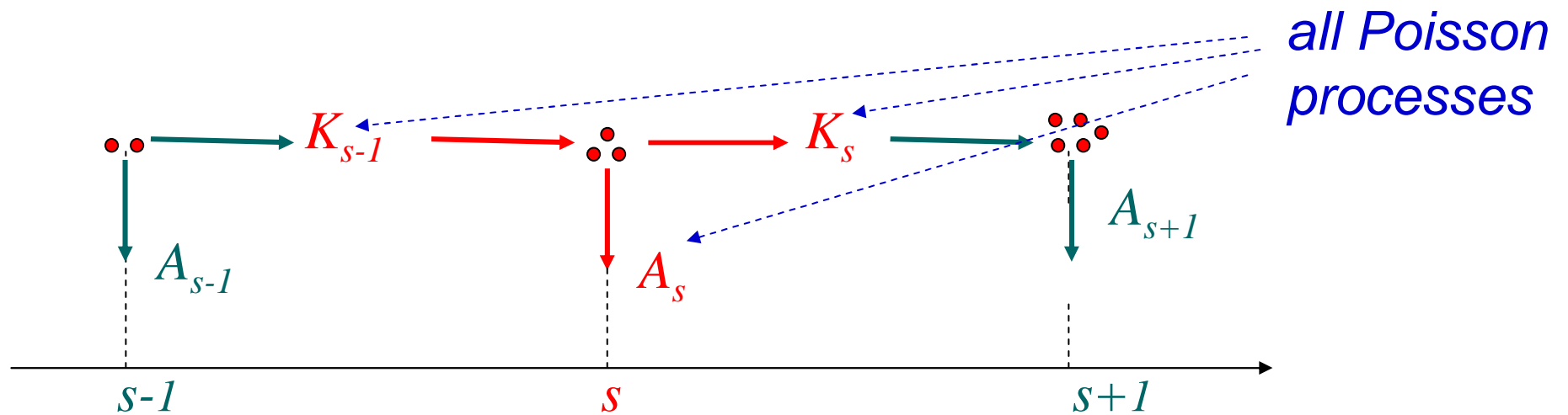
significance
of these
- next slide

$$q_s = (1 - p_s) \text{ and } \bar{q}_s = \prod_{m=0}^s q_m$$

Mathematically (contd)



Splitting K_{s-1} into A_s and K_s

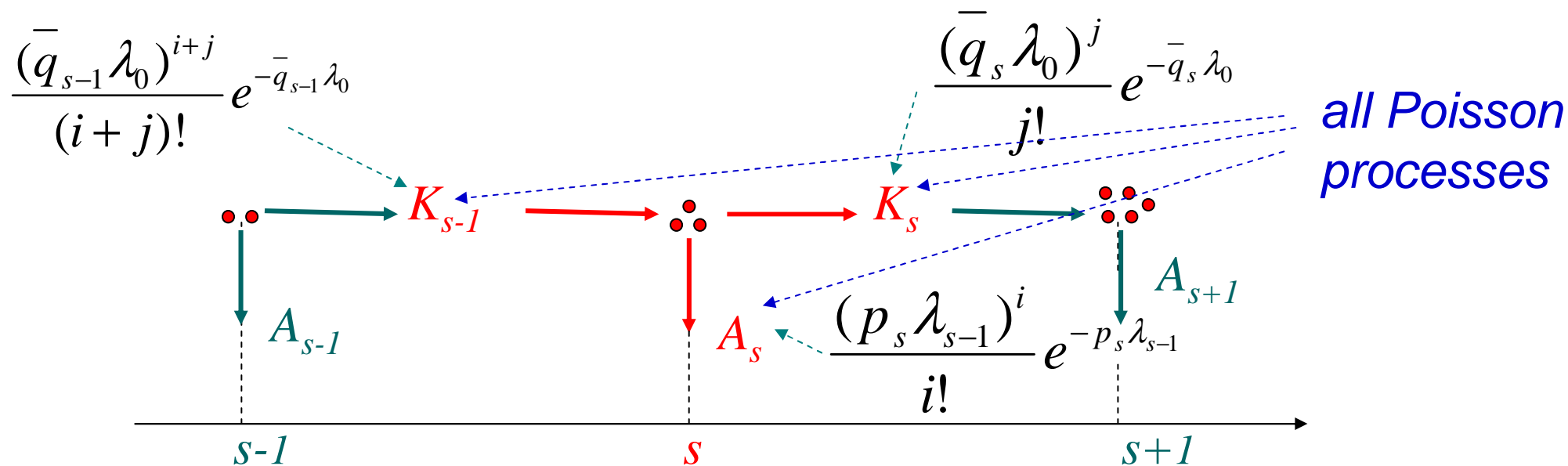




Mathematically (contd)

Splitting K_{s-1} into A_s and K_s

[expressions (in black) giving probabilities of K_{s-1} taking value $i+j$, A_s taking value i and K_s taking value j]



Can calculate timeout failure rate, buffer occupancy, etc.



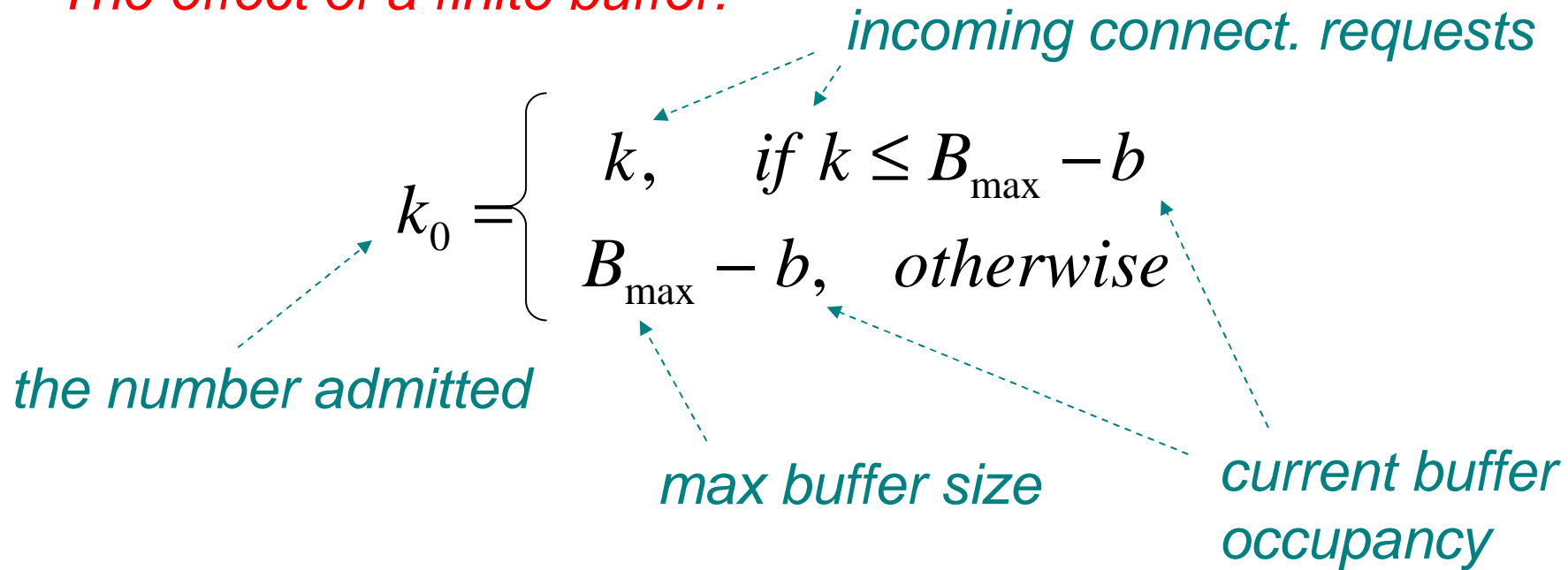
What does it give ... (and does not)

- **Failures** are a **Poisson process** with a rate $\bar{q}_{T_{out}} \lambda_0$
 - request rate λ_0 reduced by a factor $\bar{q}_{T_{out}}$ to give the failure rate
- Knowing P_{K_s} , it is possible to:
 - Obtain **statistics of half-open connections**
 - **Detect overcrowding** of the backlog queue
- Provides a **'benchmark'** for checking simulation results (in addition to a useful insight)
- Poisson distribution is easier, **other distributions are harder**
- **Limitations:**
 - A **restricted model** (incoming traffic distribution, infinite buffer, etc.)
 - **Attacker** statistical models **need to be superimposed** on the legitimate traffic model (for both attack SYN traffic and attack RTT distributions).
 - **Poisson process** is a reasonable representation for legitimate traffic, but **not for attack traffic**.

The General Case by Simulation (with **Finite Buffer**)



The effect of a finite buffer:



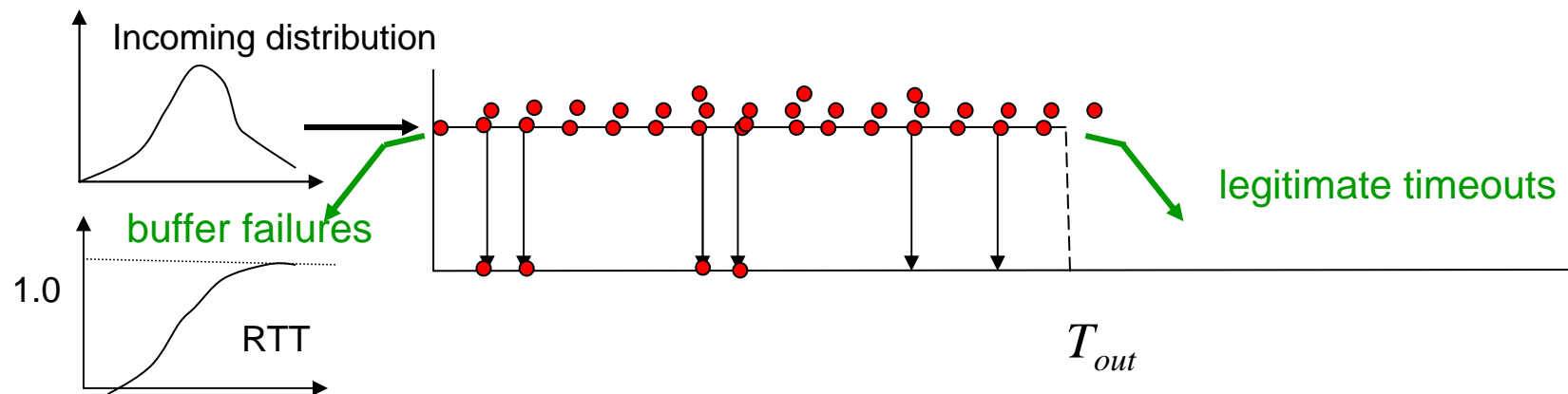
Buffer (backlog queue) failures:

$$k - k_0$$

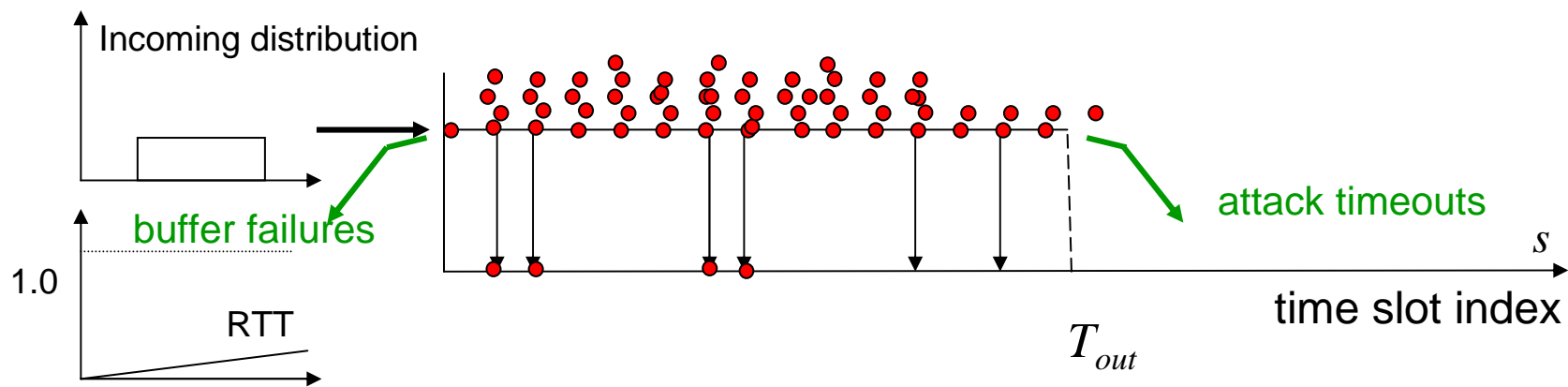


The General Case by Simulation (with **Finite Buffer**)

Consider two traffic streams: a) Legitimate traffic



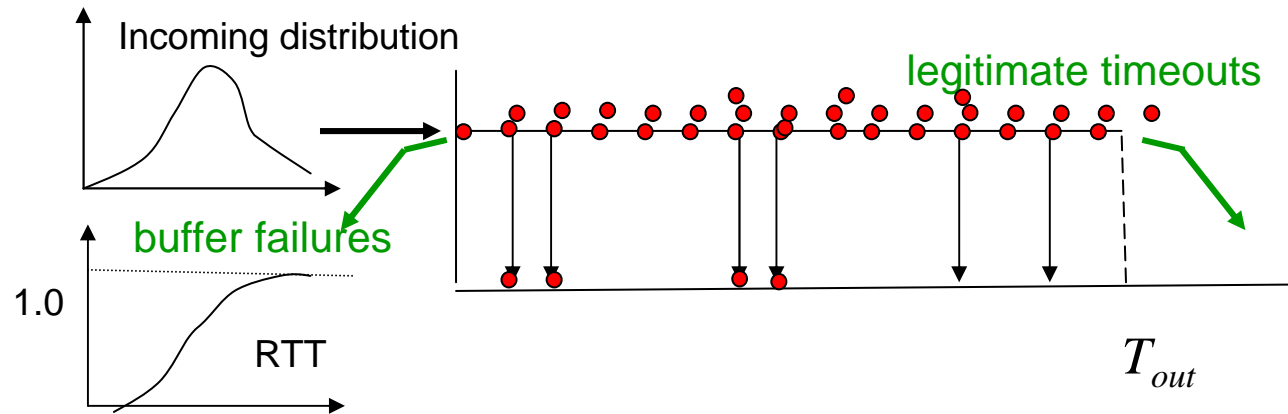
b) Attack traffic



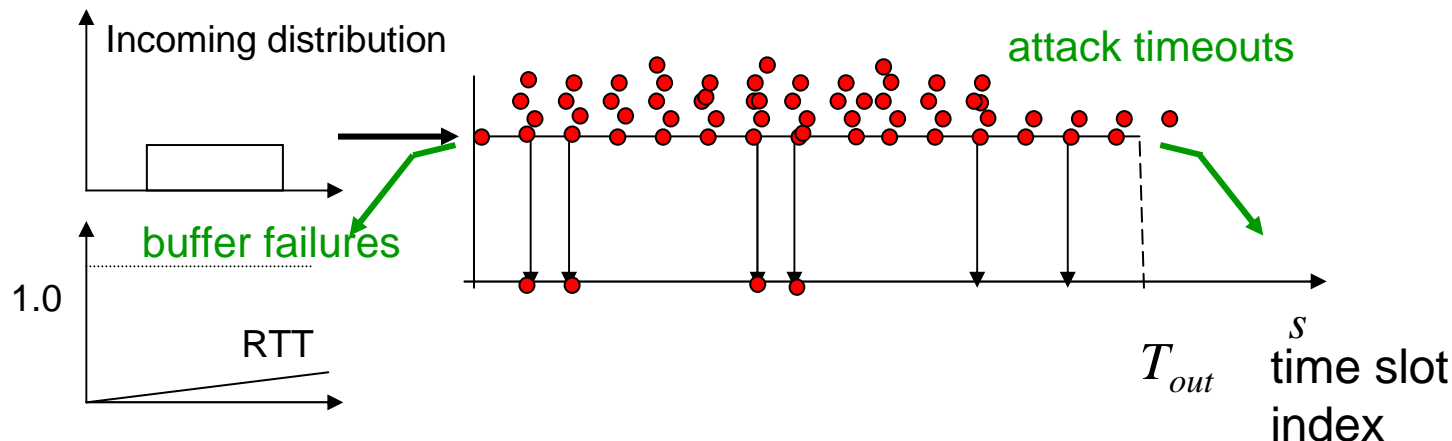
The General Case by Simulation (with **Finite Buffer**)



Consider two traffic streams: a) Legitimate traffic



b) Attack traffic



Simulation algorithm

Starting from the back end establish unacknowledged messages and shift them to the right

Record any time-out failures

Upon reaching the front end admit a new batch of incoming messages

Record any buffer failures



An Example: RTT Distributions of Normal Traffic

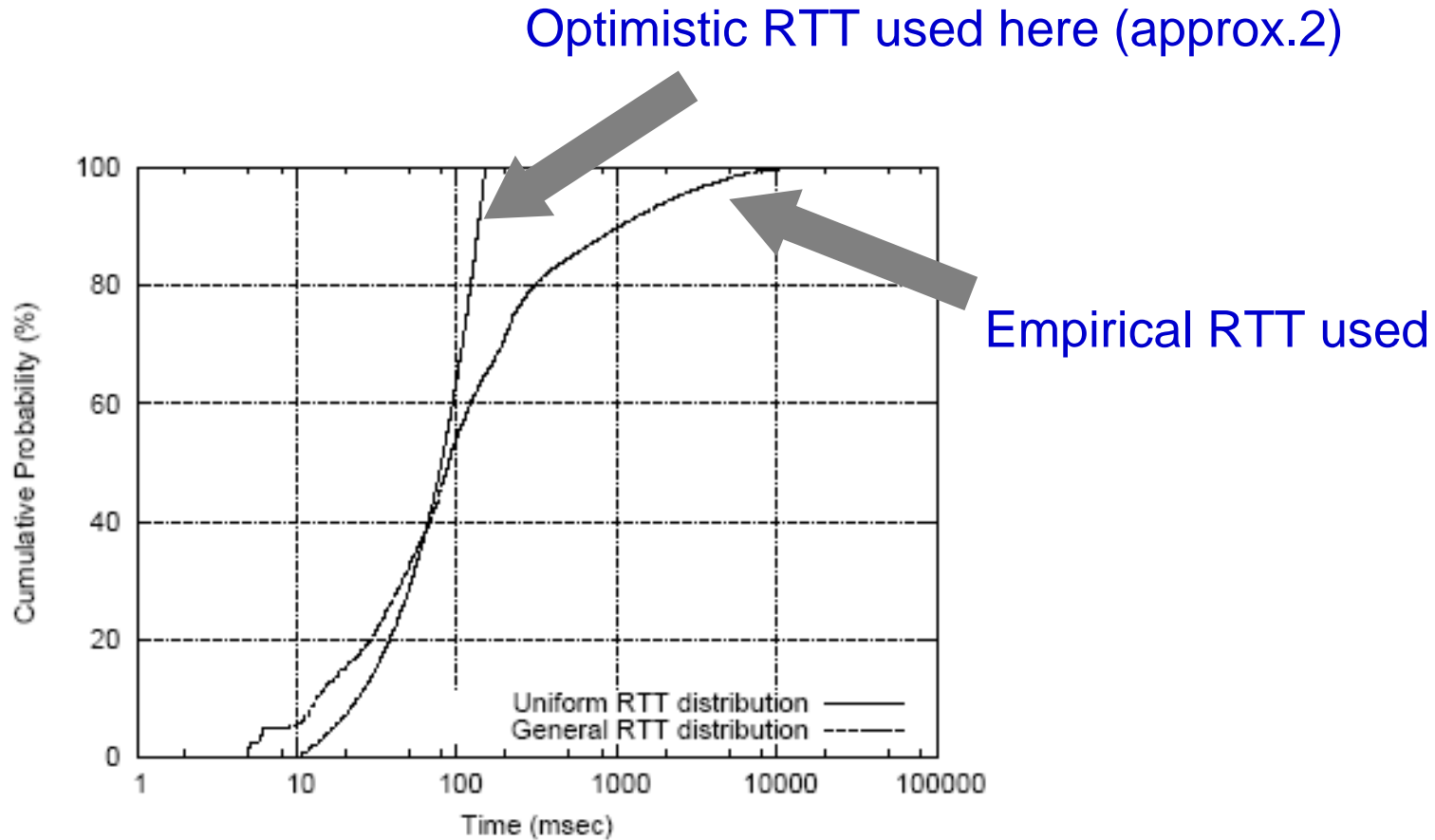
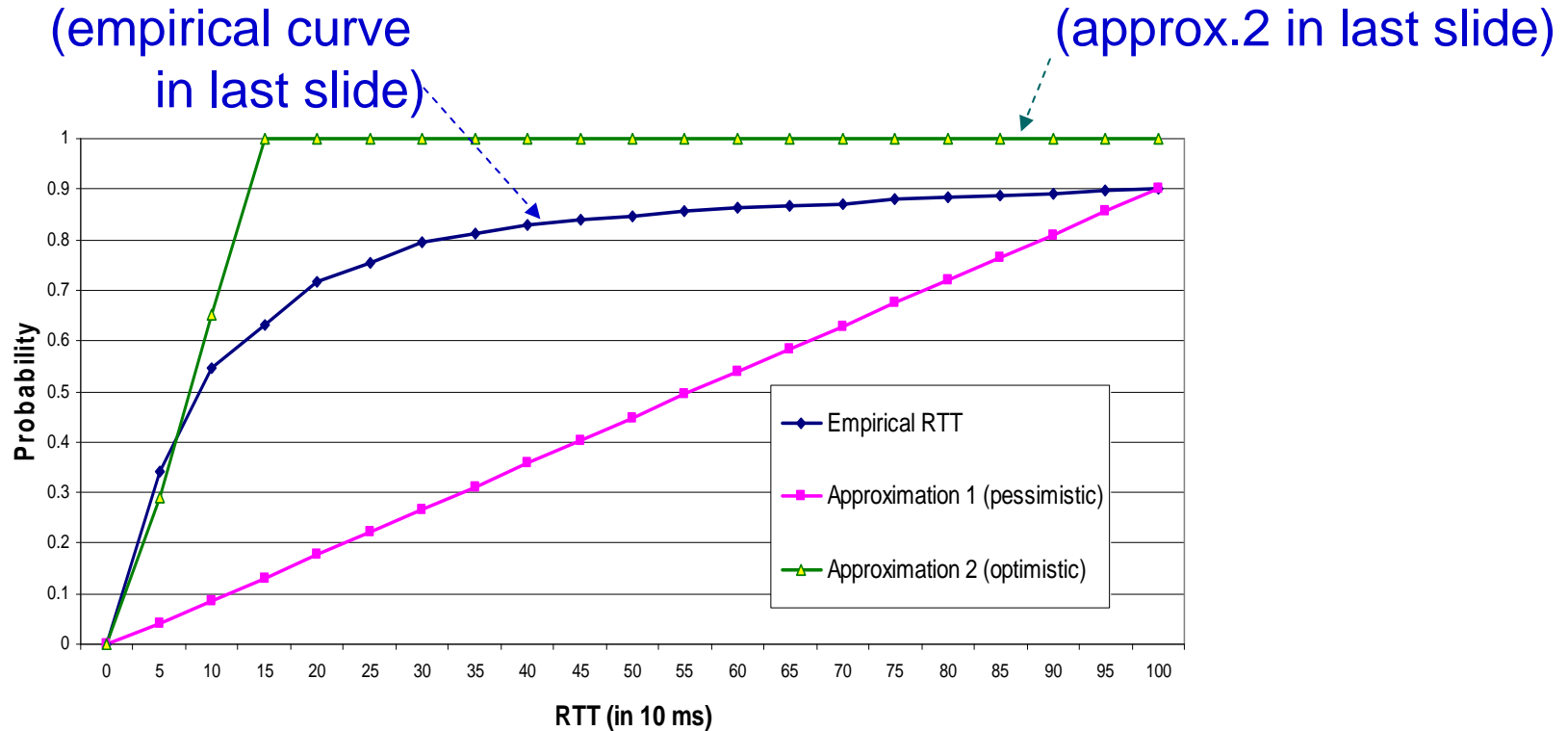


Figure 3.6: CDF of the general RTT distribution

RTT of normal traffic

[Source: N. Le, PhD thesis,
University of North Carolina, 2005]

An Example: RTT Distributions of Normal Traffic



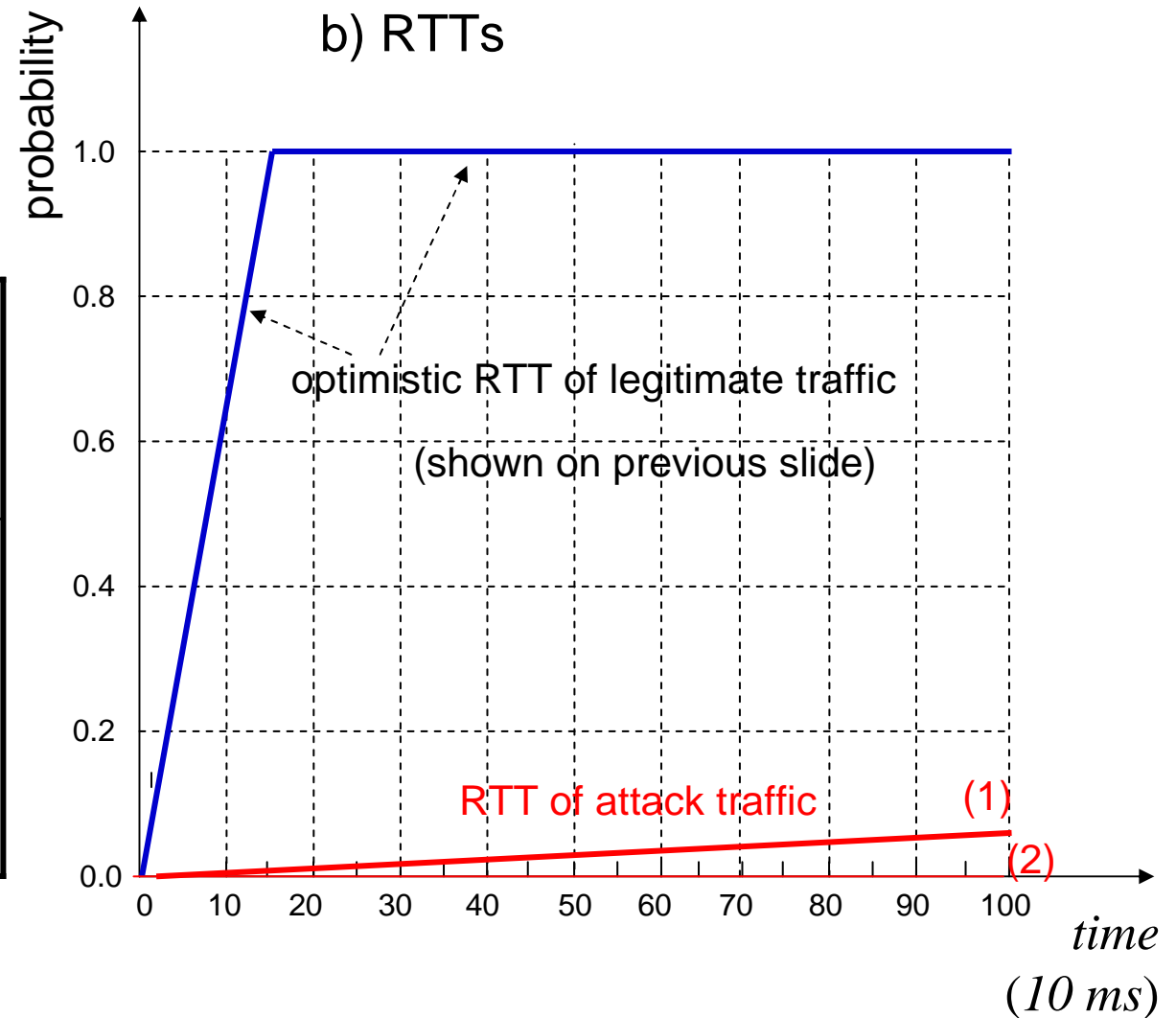
RTT of normal traffic (approximations used)

An Example: Environmental (Traffic) Parameters



a) Incoming traffic:

<p>Legitimate traffic: Poisson (others give similar results due to high rate)</p>	$\lambda = 50$
<p>Attack traffic: discrete uniform</p> $P_N(x) = \begin{cases} \frac{1}{N+1} & \text{for } x=0, \dots, N \\ 0 & \text{for } x > N \end{cases}$	<p>N= 100, 300, 500, 700 (mean rate = N/2)</p>



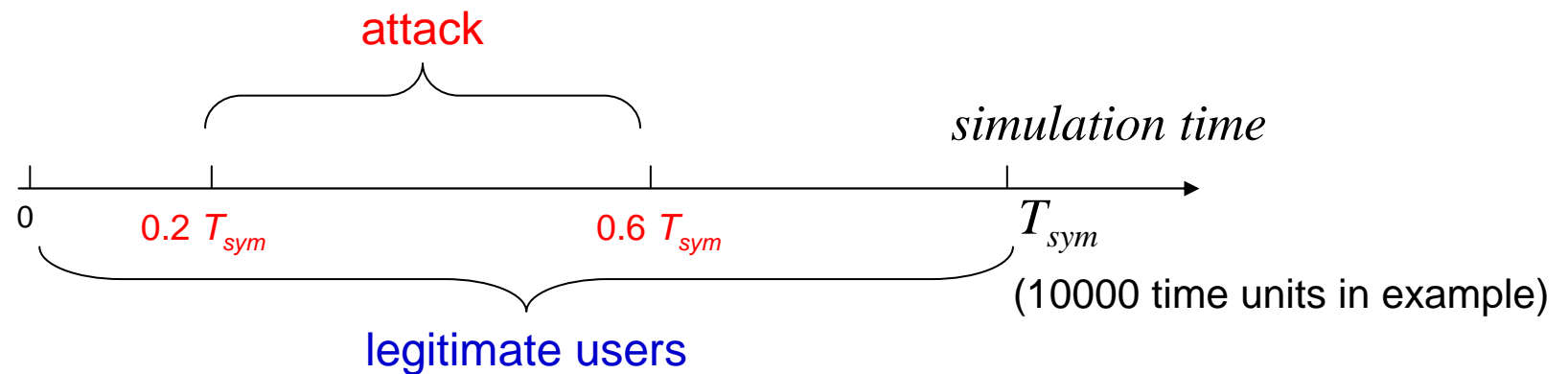
An Example: System Parameters



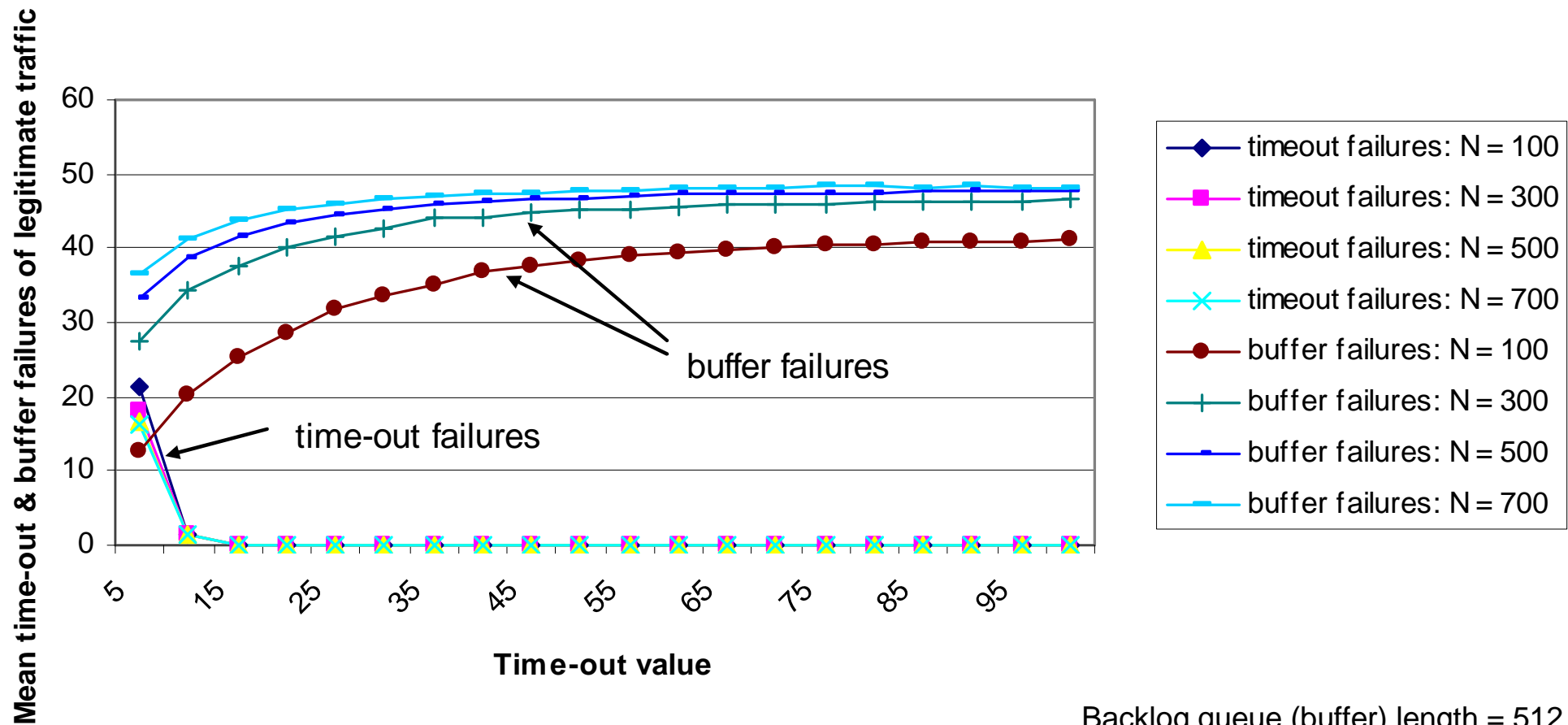
System parameters:

TCP Timeout settings	$T_{\text{out}}: 5, \dots, 100 \text{ ms}$
Backlog queue (buffer) length	$B_{\text{max}}: 512, 1024, 2048$

Simulation:

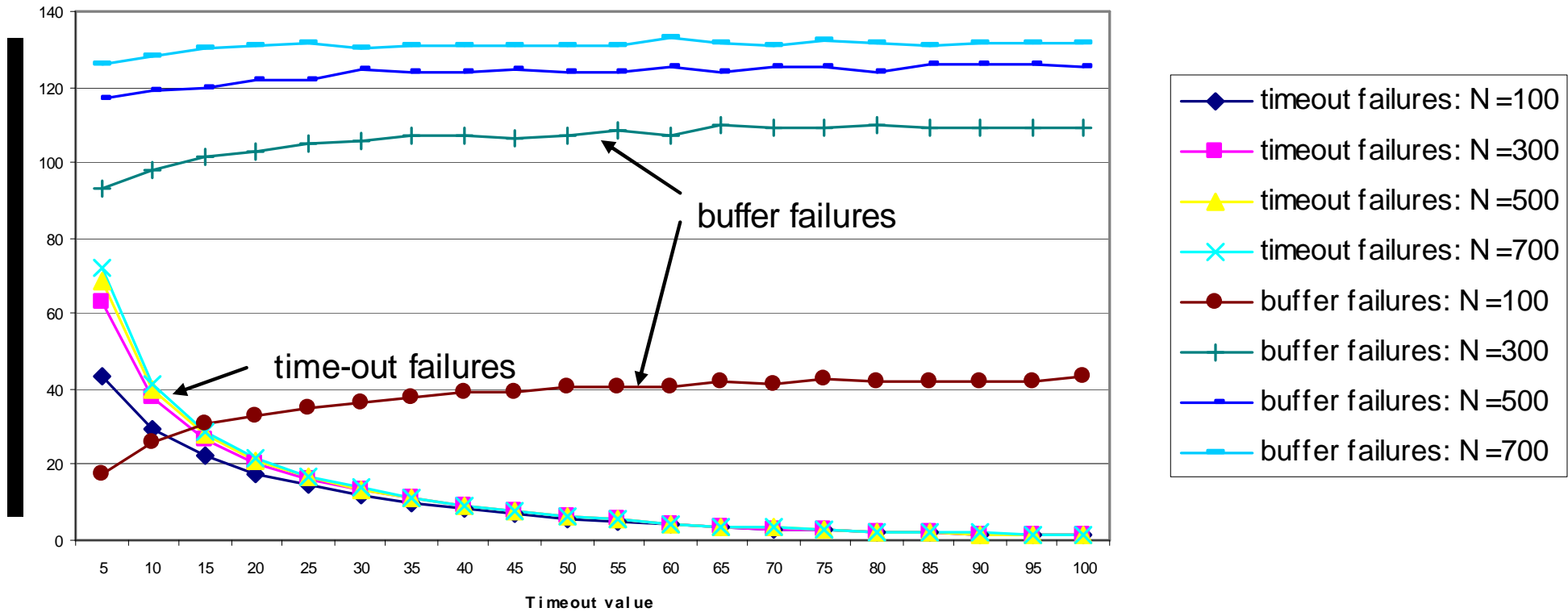
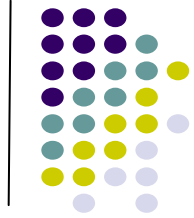


Legitimate Traffic Failures

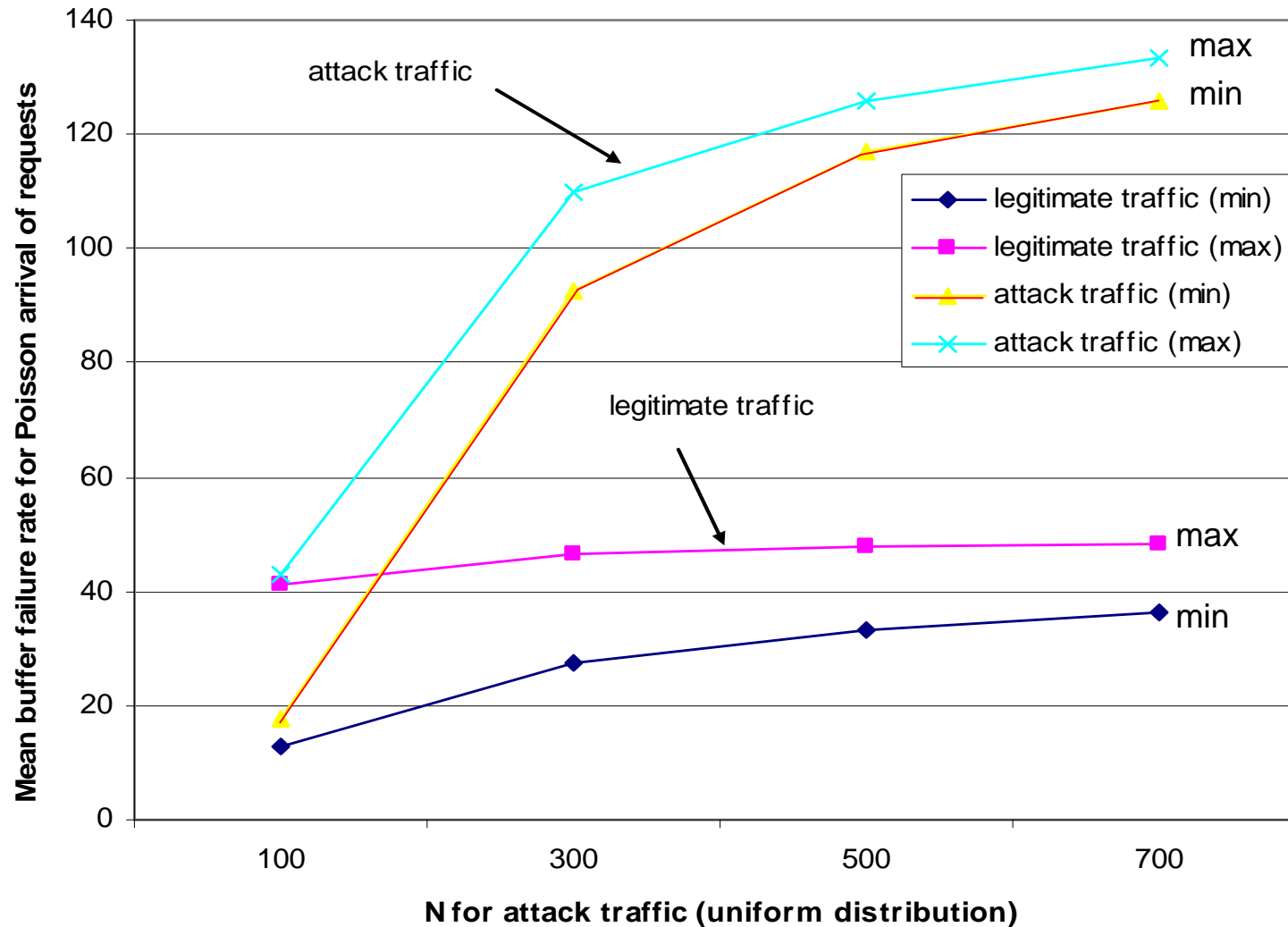


Backlog queue (buffer) length = 512

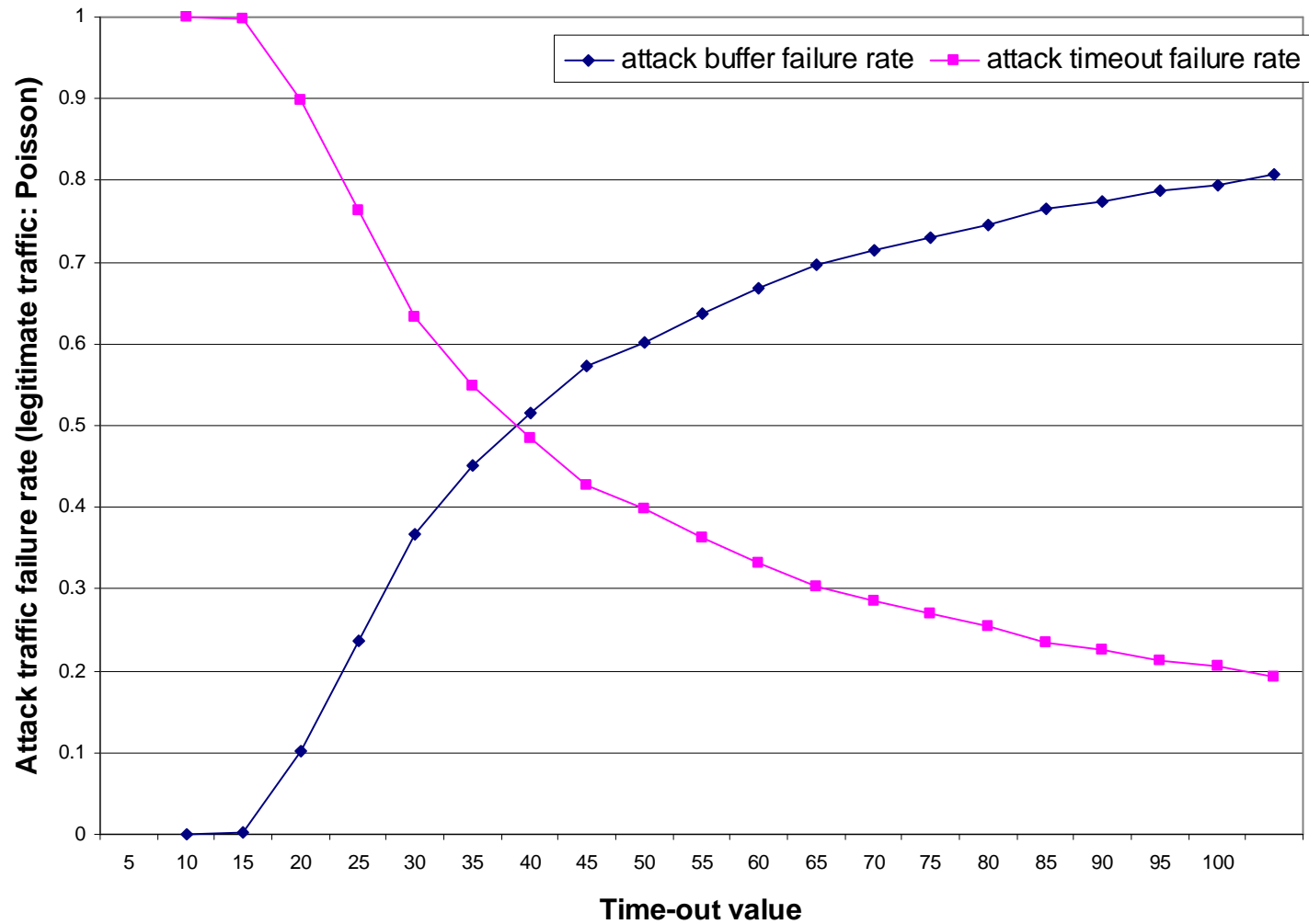
Attack Traffic Failures



Buffer Failure Rate vs. Attack Traffic Intensity



DDoS Attack with All Spoofed Addresses





What does it give ... (and does not)

- **To detect attacks**, use rate of SYN traffic in conjunction with **sudden changes** in
 - Timeout failures
 - Buffer failures
- It helps **avoiding** false identification of ‘**flash crowds**’ as DoS attacks.
- **To combat attacks**, the aim is to
 - **Adjust system parameters** dynamically based on comparisons of:
 - Run-time measurements of incoming traffic intensity and failure rates
 - Predicted variation of failure rates based on incoming traffic models
 - Use above in conjunction with other techniques (e.g. **proxy defence nodes** near victim site).
- **Accurate** predictions require:
 - **Realistic traffic models** matching empirical observations (especially for SYN traffic)
 - **Accurate RTT distributions** for SYN/ACK and ACK traffic
 - Efficient algorithms to deal with **4-5 order difference** between maximum and minimum RTT values

Conclusions



- An approach to detecting and taking corrective actions against DoS attacks.
- A combined strategy based on
 - Statistical characterisation of incoming SYN traffic and Round Trip Time (RTT)
 - Buffer (backlog queue) and time-out failures
- Exceptional reliance on RTT limits any adverse effect to legitimate users with high RTT values
- The model is parameterised with respect to the distribution function of incoming traffic:
 - Possible to establish the worst possible request patterns through sensitivity analysis
 - Especially useful in the context of scarce definitive information on Web-based traffic

Thank you!

