ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Network Protocol Design and Evaluation

08 - Analytical Evaluation

Stefan Rührup

University of Freiburg
Computer Networks and Telematics

Summer 2009

CoNe Freiburg

IIF
INSTITUT FÜR INFORMATIK FREIBURG

Overview

- **In the last chapter:**
  - Simulation

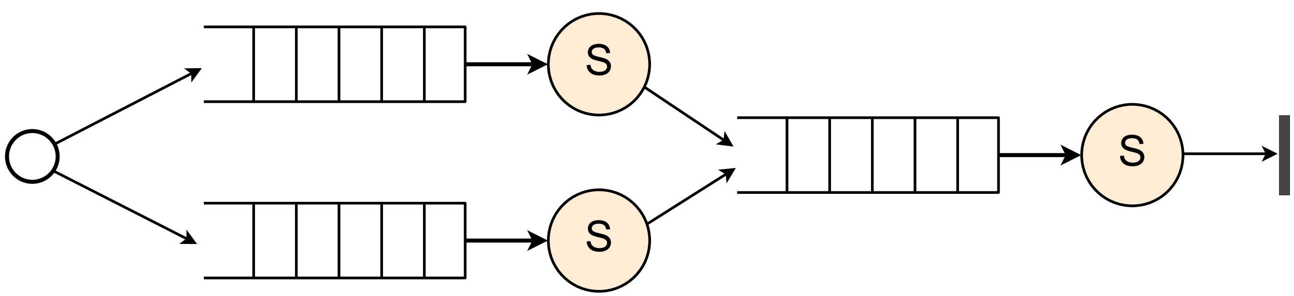
- **In this part:**
  - Analytical Evaluation: case studies

Analytical Evaluation

- **Analytical validation**
  Proof of correctness, deadlock-freedom etc.
  (cf. Chapter 5 on Validation)
- **Analytical performance evaluation**
  - Requires model abstraction
  - Methods of distributed system analysis,
    esp. Queuing theory

Queuing models (1)

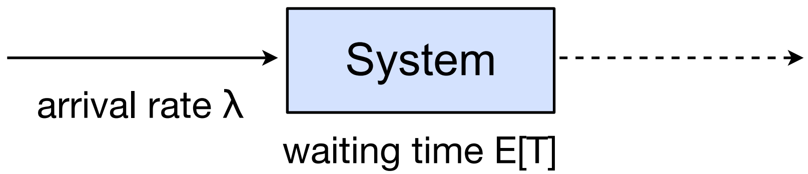
- System description by processes with focus on task arrival, queuing, processing
- Load generation and service times described by stochastic processes (e.g. Poisson process)
- Analytical performance measures can be determined



Example of a queuing network

Queuing Models (2)

- **Little's Law**: The long-term average number of tasks in a system E[X] equals the product of long-term average arrival rate λ and average waiting time E[T]: $E[X] = \lambda E[T]$



- Arrival and service times are described by stochastic processes (cf. Renewal processes in Chapter 7)

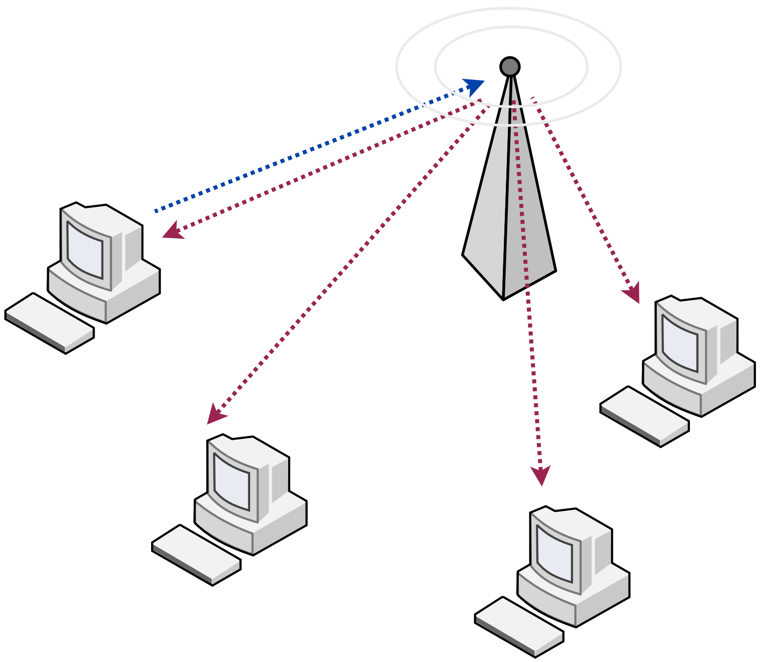
Case Study 1: Analysis of ALOHA

- ALOHANET: Wireless packet radio network with star/broadcast topology
- 2 channels: Messages are sent by hosts to the hub station using the inbound channel. The hub brodcasts the message to all stations using the outbound channel (message delivery and feedback to the sender).
- The ALOHA Protocol
  - Whenever you have data, send it
  - If there is a collision, try to retransmit

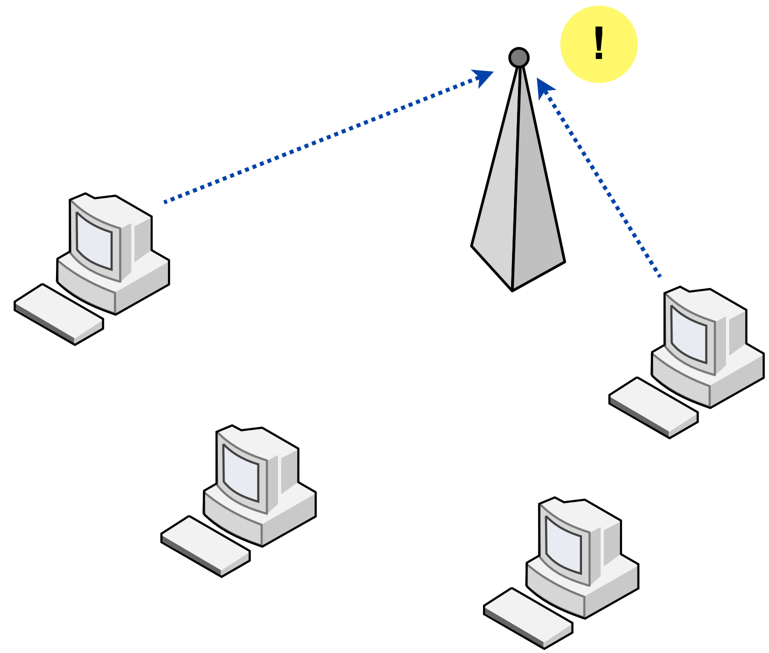
ALOHAnet

- Wireless packet radio network with star/broadcast topology
- 2 channels: Messages are sent by hosts to the hub station using the inbound channel. The hub brodcasts the message to all stations using the outbound channel (message delivery and feedback to the sender).
- The ALOHA Protocol
  - Whenever you have data, send it
  - If there is a collision, try to retransmit

ALOHA



Transmission and Re-Broadcast



Collision

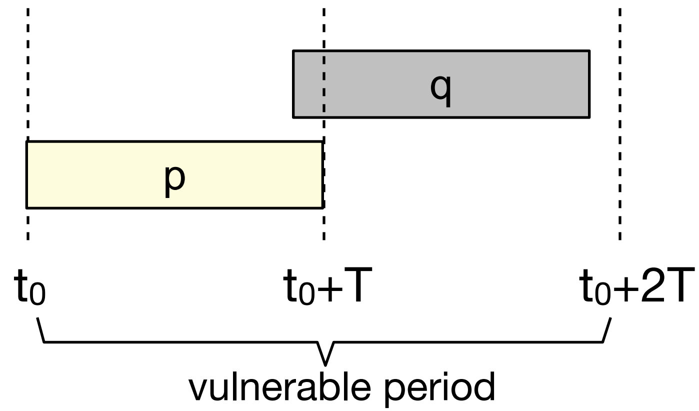
Throughput Analysis (1)

- Assumptions
  - Number of stations: N
  - Packet transmission time: T
  - Each station transmits with probability p per time interval T
  - Packet injection follows a Poisson process with arrival rate $\lambda = Np$ (arrivals at the hub station within T).

- *Metric*: Throughput = number of successfully delivered packets per time interval.

Throughput Analysis (2)

- Collisions: Packets can collide with others within a time interval of 2T (*vulnerable period*)



Throughput Analysis (3)

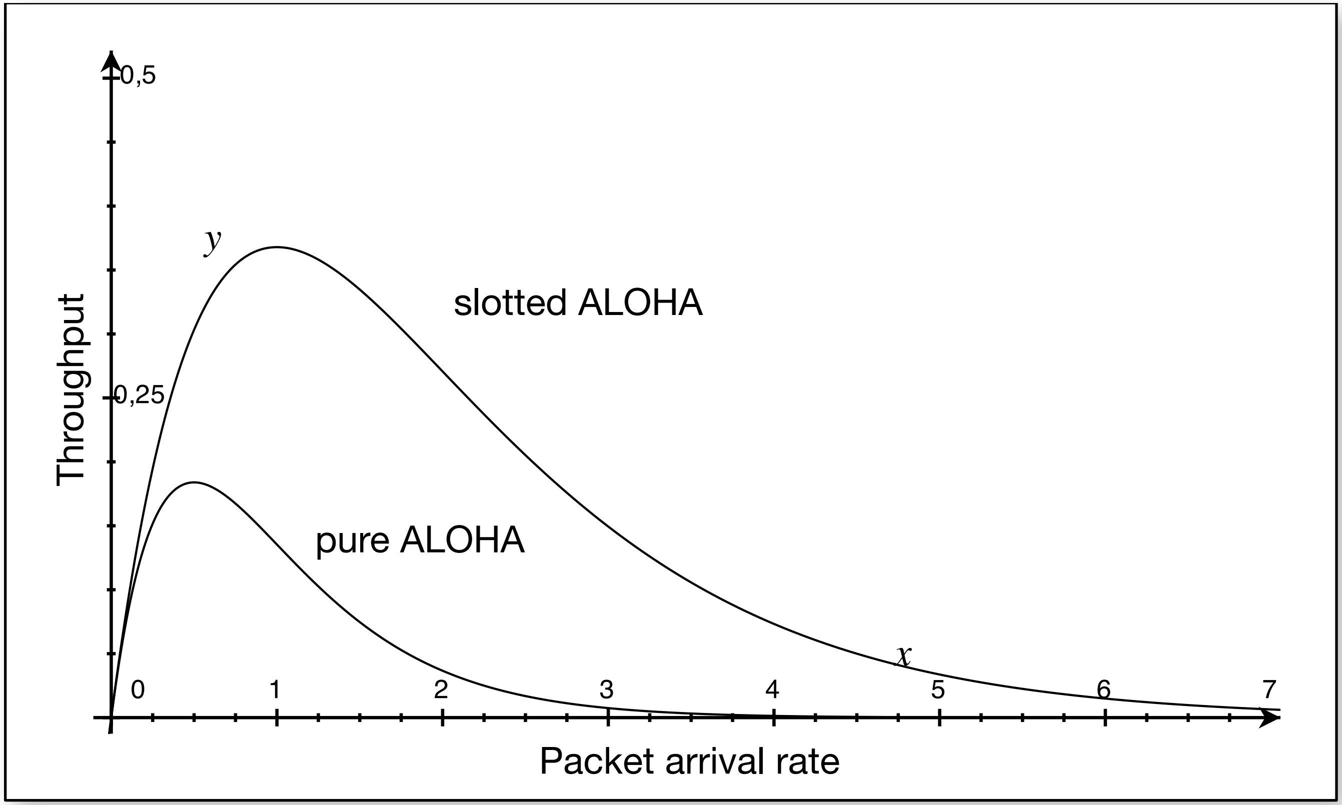
- We calculate the probabilities according to the Poisson distribution:
  - Pr[Success] = Pr[no other transmissions within 2T] $= P(0,2) = e^{-2\lambda} = e^{-2Np}$
  - Throughput = Mean number of arrivals * Pr[Success] $= \lambda e^{-2\lambda} = Np\, e^{-2Np}$
  - Maximum: $e^{-2\lambda} - 2\lambda\, e^{-2\lambda} = 0$ when $\lambda = \frac{1}{2}$
    Optimal throughput = $1/2\, e^{-1} \approx 0.18$

Poisson distribution: $P(k,t) = Pr_{\lambda,t}[X = k] = \frac{\lambda^k}{k!}\, e^{-\lambda t}$

Throughput Analysis (4)

- **Slotted ALOHA:** Transmissions are synchronized and begin at time slots of length T. Thus, the vulnerable period is reduced to T.
- Analysis:
  - Pr[Success] = P(0,2) = $e^{-\lambda} = e^{-Np}$
  - Throughput = $\lambda e^{-\lambda} = Np\, e^{-Np}$
  - Maximum is reached at $\lambda=1$ with a throughput of $1/e \approx 0.3679$

Throughput Analysis (5)



Backlogged Packets (1)

- What we did not consider so far: There are backlogged packets after a collision, which will be retransmitted with probability r.
- Assume that there are M (out of N) stations with backlogged packets. Then the expected number of transmission attempts is

  $$\lambda(M) = (N - M)a + Mr$$

  where $a = 1\text{-}e^{-\lambda/N}$ is the arrival probability *per station*.
- P[Success] = P[one new packet and no backlogged packet *or* no new packet and one backlogged packet]
  $= (N\text{-}M)\ a\ (1\text{-}a)^{N-M-1}\ (1\text{-}r)^{M} + (1\text{-}a)^{N-M}\ M(1\text{-}r)^{M-1}r.$

[Barbeau, Kranakis: Principles of Ad-hoc Networking, Wiley, 2008]

Backlogged Packets (2)

- $P[\text{Success}] = (N-M)\, a\, (1-a)^{N-M-1} (1-r)^M + (1-a)^{N-M} M (1-r)^{M-1} r.$
- We use $x/(1-x) \approx x$ and write

$$(N-M)\, a\, (1-a)^{N-M-1} (1-r)^M = (N-M)\, a\, (1-a)^{N-M} (1-r)^M / (1-a)$$
$$\approx (N-M)\, a\, (1-a)^{N-M} (1-r)^M$$
$$(1-a)^{N-M} M\, (1-r)^{M-1}\, r = (1-a)^{N-M} M (1-r)^M\, r / (1-r)$$
$$\approx (1-a)^{N-M} M (1-r)^M\, r$$

- $P[\text{Success}] = (N-M)\, a\, (1-a)^{N-M} (1-r)^M + (1-a)^{N-M} M\, (1-r)^M\, r$

$$= \left((1-a)^{N-M} (1-r)^M \right)\left((N-M)\, a + M\, r \right)$$

- We use $(1-x)^y \approx e^{-xy}$ and get $(1-a)^{N-M} (1-r)^M \approx e^{-a(N-M)}\, e^{-Mr}$
- $P[\text{Success}] \approx e^{-(a(N-M)+Mr)} \left((N-M)\, a + M\, r \right) = e^{-\lambda(M)}\, \lambda(M)$

[Barbeau, Kranakis: Principles of Ad-hoc Networking, Wiley, 2008]

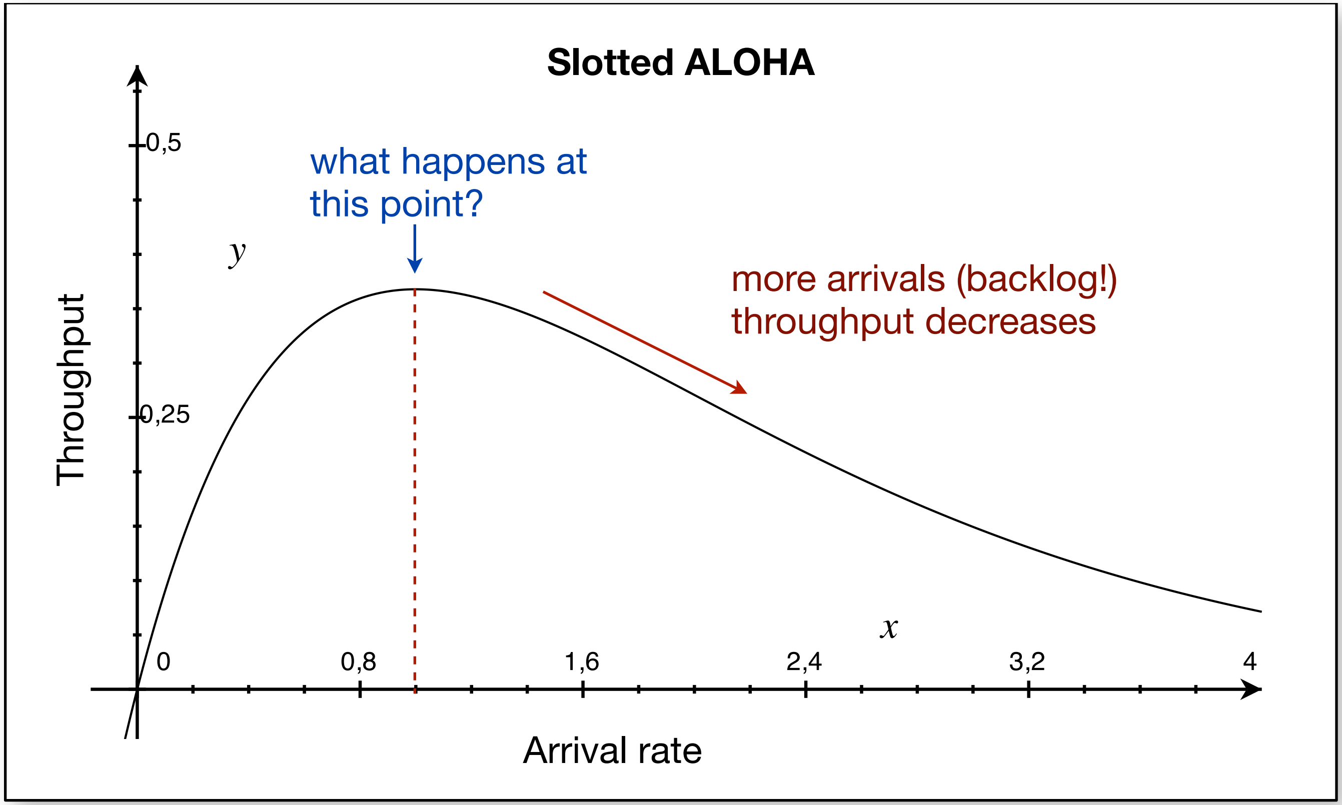
Backlogged Packets (3)

- P[Success] $\approx e^{-(a(N-M)+Mr)}$ ((N-M) a + M r) = $e^{-\lambda(M)}\ \lambda(M)$
- Thus we can approximate the additional arrival of backlogged packets by a Poisson process with mean $\lambda(M)$
- The throughput is maximal if $\lambda(M) = (N - M)a + Mr = 1$
- Then the retransmission probability is

$$r = 1/M - a(N-M)/M = 1/M - (1-e^{-\lambda/N})(N-M)/M$$
$$= (1-M-N)/M + (1-e^{-\lambda/N})(N-M)/M$$

[Barbeau, Kranakis: Principles of Ad-hoc Networking, Wiley, 2008]

On the Stability



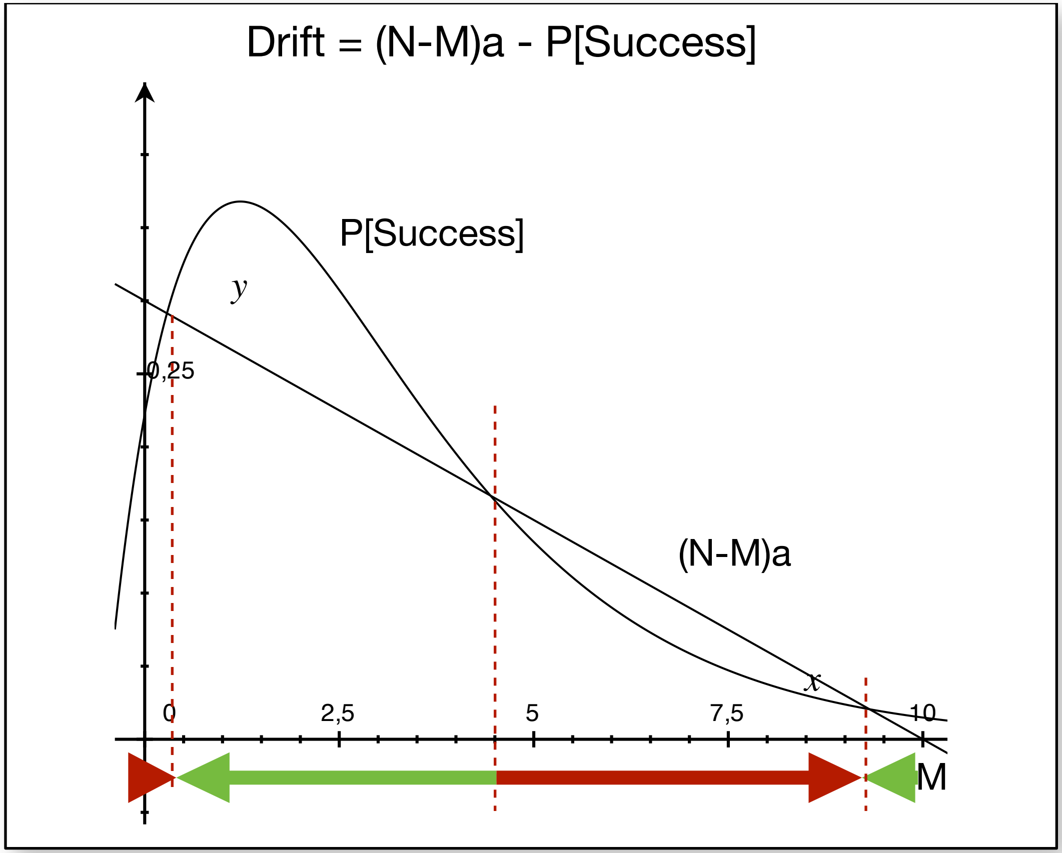
System state and Drift

- **System state:** number of stations with backlogged packets
- **Drift:** Change of backlogged stations per slot time

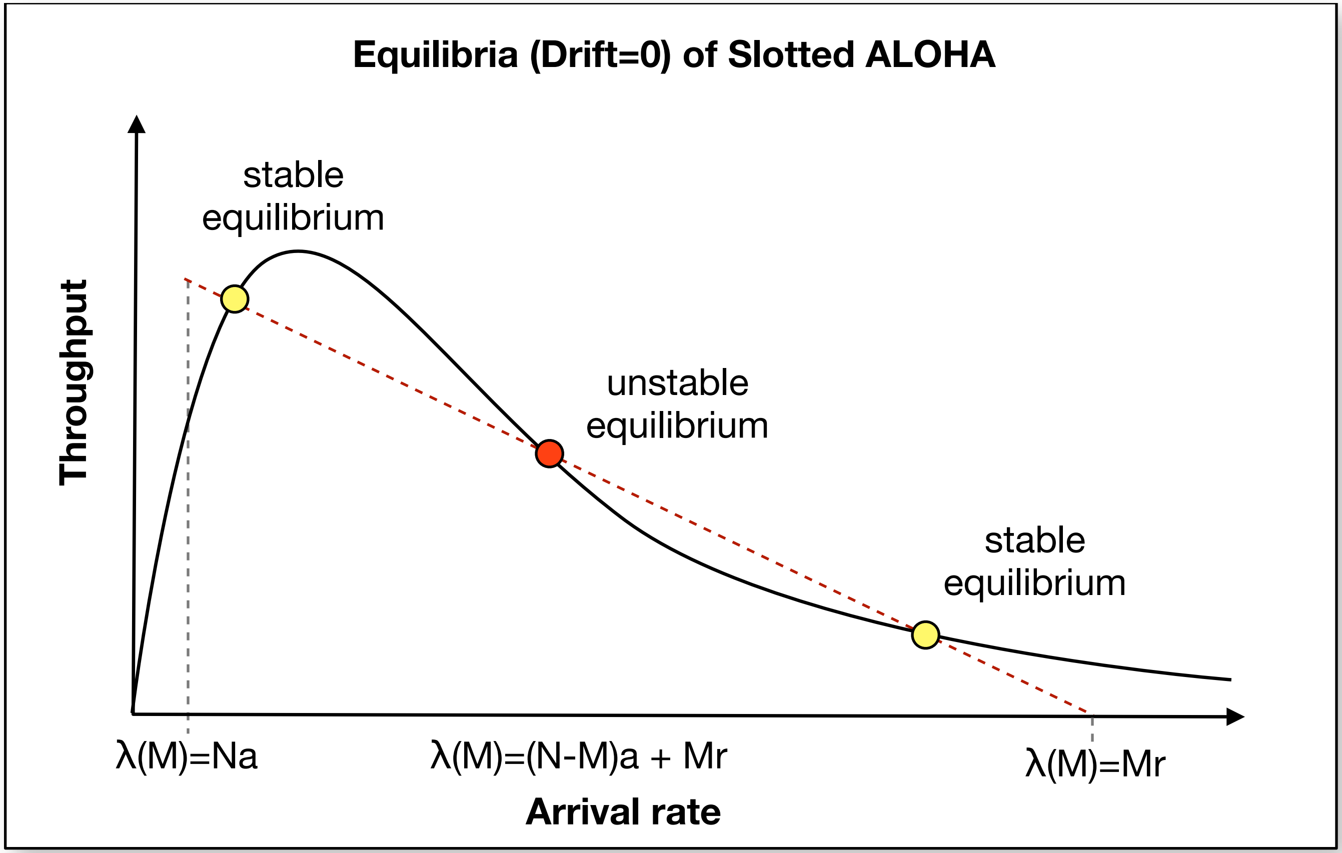
  $$D_M = (N-M)\, a - P[\text{Success}]$$

  (Difference between newly arriving packets and probably a sent packet)
- The drift indicates the direction in which the system state changes

Drift



Arrival rate and stability



Parameter settings

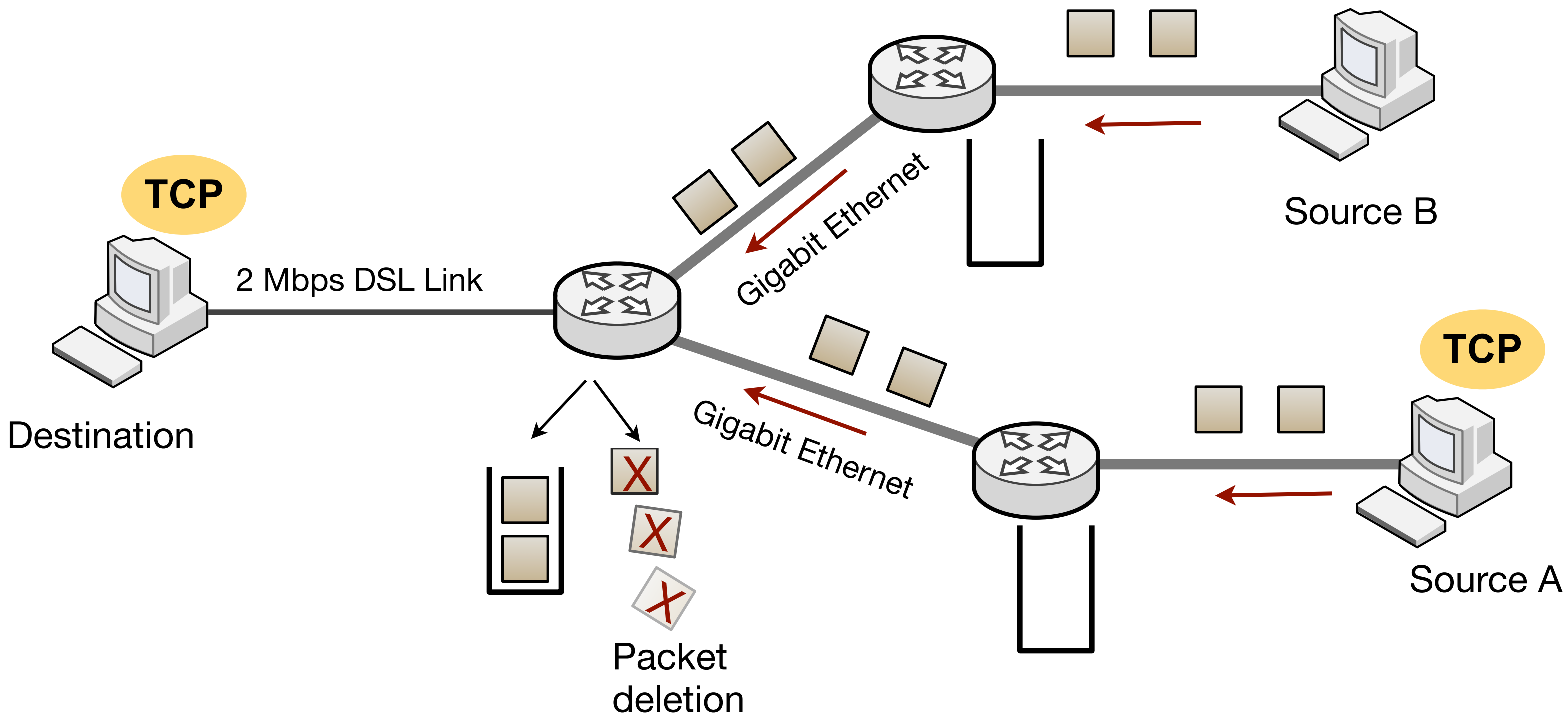
- Increasing the retransmission probability r:
  - Backlogged packets are reduced, but the unstable equilibrium can be exceeded quickly
- Reducing r increases the delay.
- There are algorithms to ensure stability
- Practically, we should keep the arrival rate below the maxium

Case Study 2: Analysis of TCP’s Congestion Control

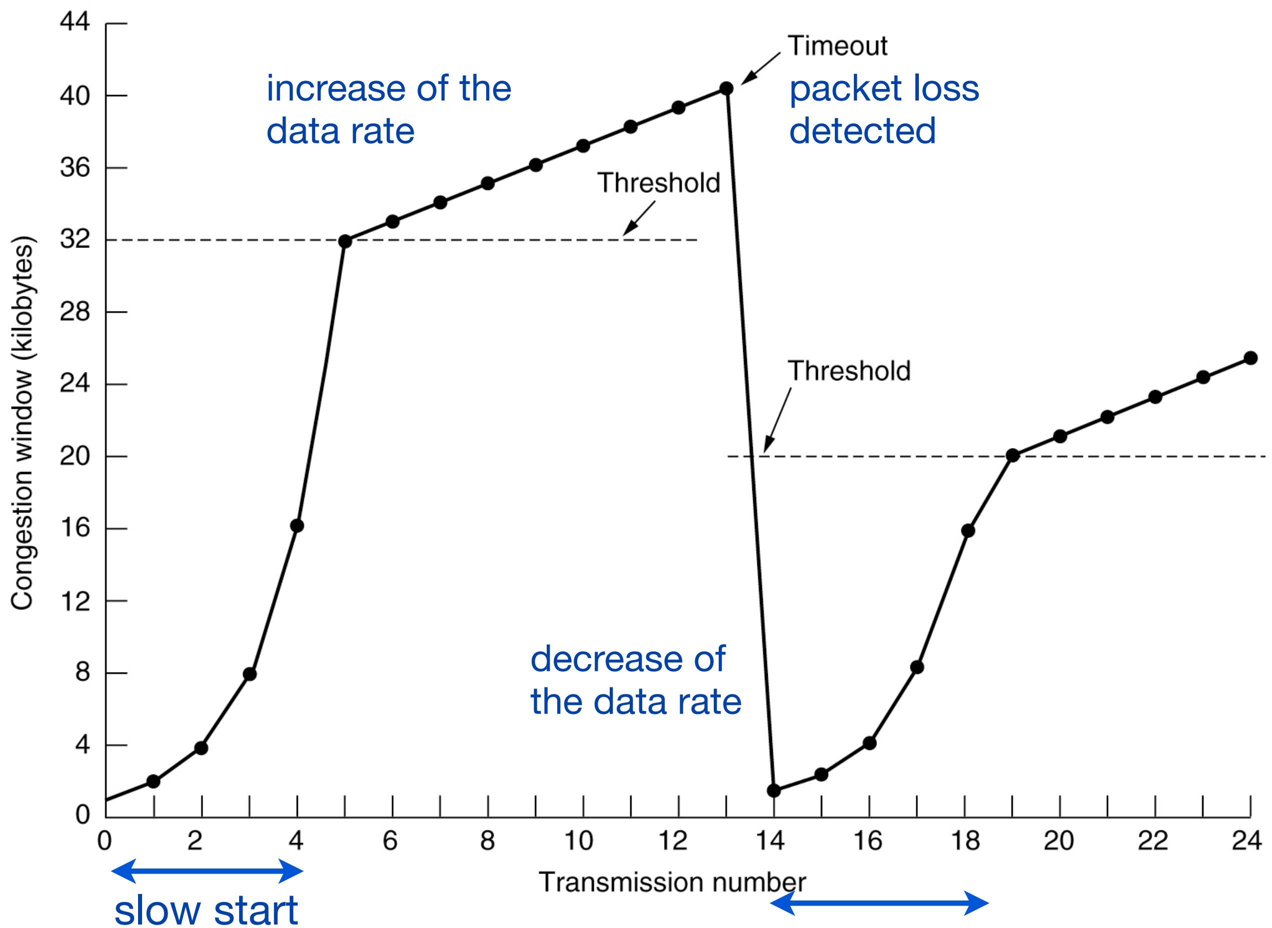
- TCP provides an acknowledged end-to-end datagram delivery service
- It uses IP (unacknowledged, connectionless) and shares the bandwidth with other traffic
- In congestion situations, routers drop packets
- TCP reacts by adapting the injection rate.
  Recall: the only available information to detect congestion situations are acknowledgements.

Congestion revisited

- IP Routers drop packets (Random Early Discard)
- TCP has to react, e.g. lower the packet injection rate



Congestion control of TCP Tahoe



[A.S. Tanenbaum, Computer Networks, 4/e, Prentice Hall]

The Analysis

- TCP's congestion control mechanism is used by multiple participants sharing the bandwidth.
- If one user reduces the data rate, bandwidth will be available for others
- **Questions**:
  - Can this algorithm provide an efficient use of the bandwidth?
  - Are all participants treated fair?

An Analytical Model

- First, we need an abstract model for the algorithm and the environment
- **The algorithmic principle** behind TCP's congestion control:
  - Increase the data rate additively, if possible
  - Reduce the data rate by 1/2 in case of packet loss
- **Abstractions:**
  - We do not consider the slow start phase
  - We assume a round-based model (round = RTT)
  - We assume a binary feedback (packet loss yes/no)
  - The communication channel is shared and can be used up to a certain bandwidth

The AIMD Principle

- TCP uses basically the following mechanism
  to adapt the data rate x (#packets sent per RTT):
  - Initialization

    x = 1

  - If the acknowledgement for a segment arrives, perform additive increase (AI)

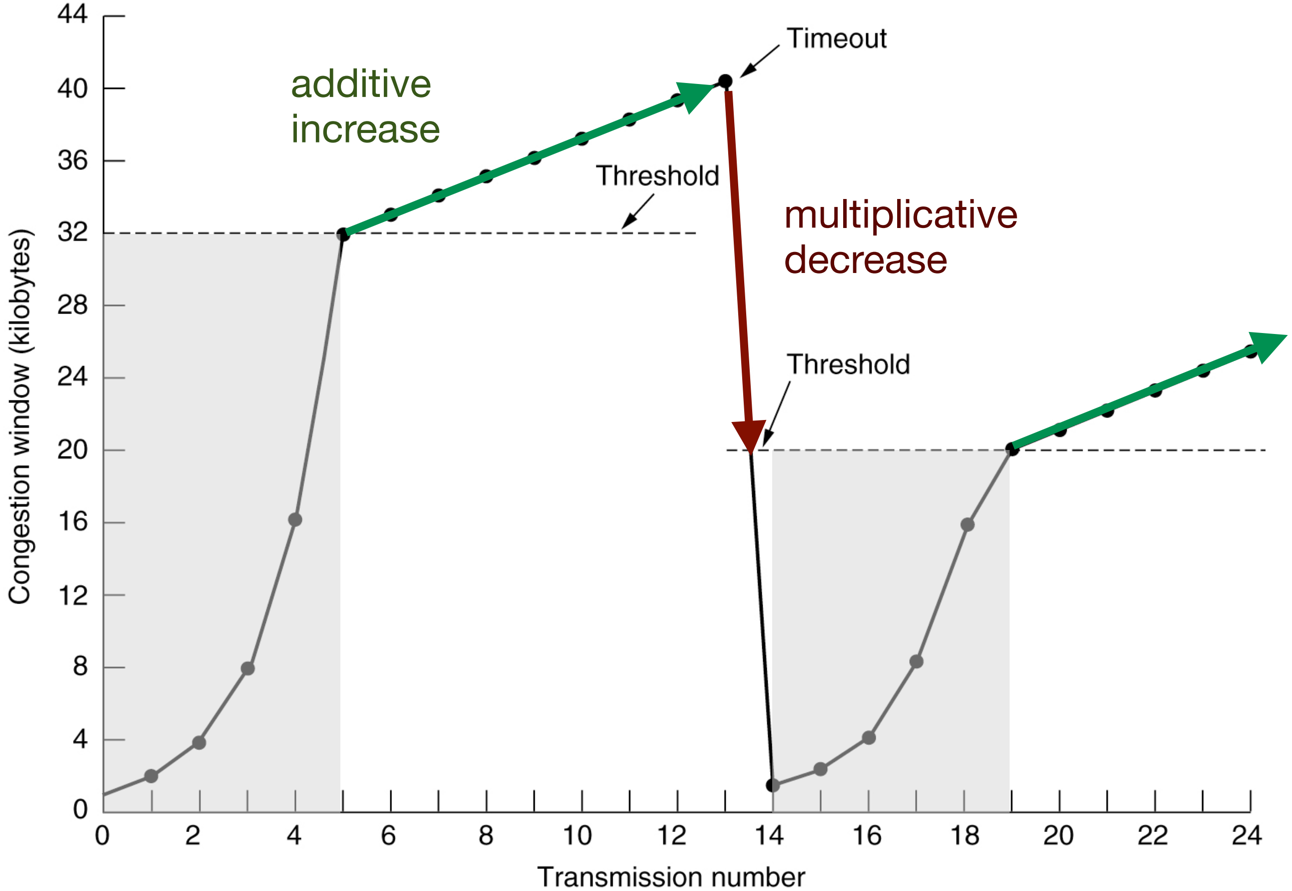
    x = x+1

  - On packet loss: multiplicative decrease (MD)

    x = x/2

[C. Schindelhauer, Algorithmische Grundlagen des Internets, Uni Paderborn, 2003]

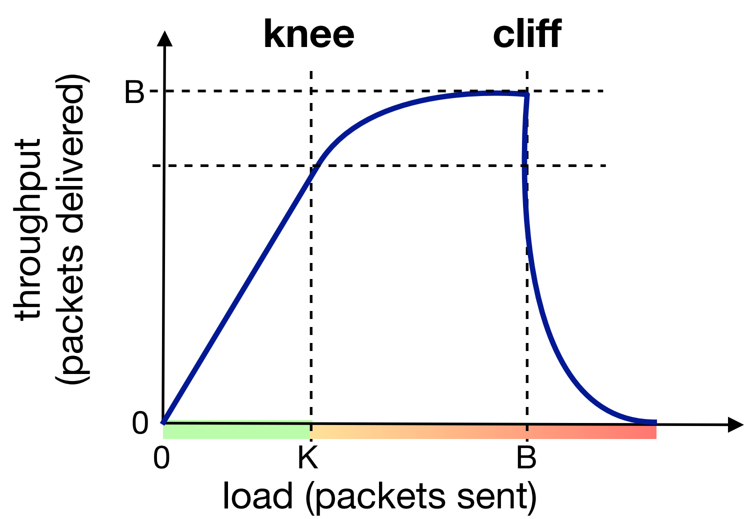
AIMD



[A.S. Tanenbaum, Computer Networks, 4/e, Prentice Hall]

Available Bandwidth

- Knee load: critical bandwidth when latency increases significantly. It is desired to keep the load around the knee.
- We assume that the timeout mechanism provides the feedback about reaching the knee load



The Data Rate Model (1)

- **Time**: round based, t = 0...
- **Participants and data rate:**
  - n participants (here called *players*)
  - participant i has a data rate of $x_i(t)$ in round t
  - overall data rate: $X(t) = \sum_{i=1}^{n} x_i(t)$
- **Feedback:**
  - feedback function y(t) (the same for all players)

$$y(t) = \begin{cases} 0, & \text{if } X(t) \leq K \\ 1, & \text{if } X(t) > K \end{cases}$$

where K is the knee load.

[C. Schindelhauer, Algorithmische Grundlagen des Internets, Uni Paderborn, 2003]

The Data Rate Model (2)

- **Data rate adaption:**
  New data rate (round t+1) is given by a function of the data rate in the past round (t) and the feedback y(t):
  $x_i(t+1) = f(\, x_i(t), y(t)\,)$
- We consider linear functions with increase and decrease parameters:

$$f(x,y) = \begin{cases} a_I + b_I x, & \text{if } y(t) = 0 \\ a_D + b_D x, & \text{if } y(t) = 1 \end{cases}$$

in case of AIMD:

$$f(x,y) = \begin{cases} a_I + x, & \text{if } y(t) = 0 \\ b_D x, & \text{if } y(t) = 1 \end{cases}$$

← is this the best choice?

[C. Schindelhauer, Algorithmische Grundlagen des Internets, Uni Paderborn, 2003]

Objective Functions

- **What is fair, what is efficient?**
- **Efficiency:** The closer to the knee load, the more efficient

  $E(x) = |X(t) - K|$ desired: $E(x) \to 0$

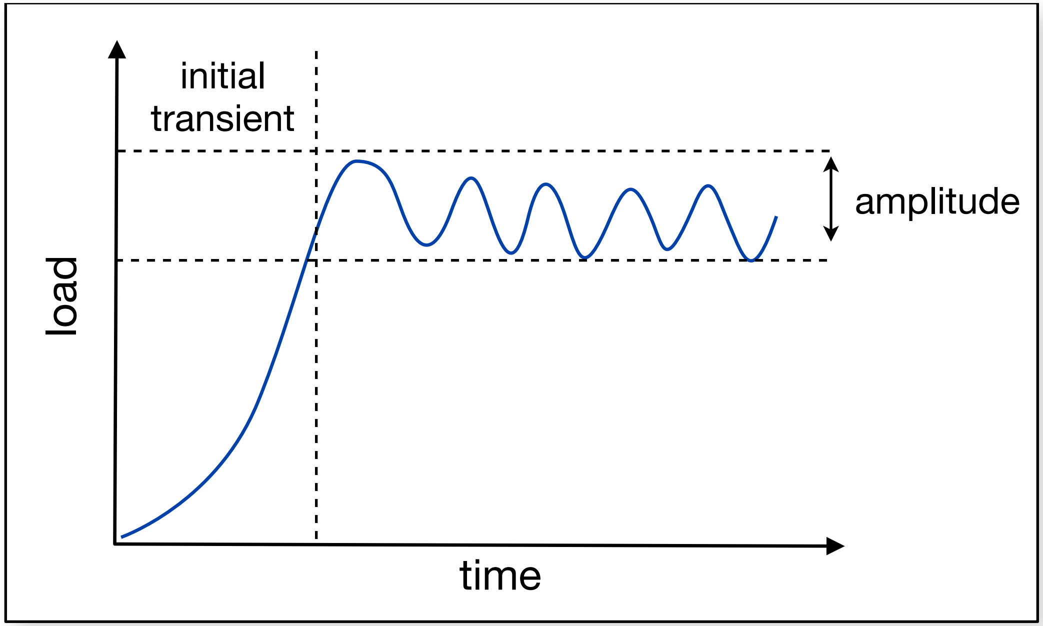
- **Fairness:** Scale-independent function with $F(x)=1$ for absolute fair situation.

$$F(x) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \sum_{i=1}^{n} (x_i)^2}$$

[C. Schindelhauer, Algorithmische Grundlagen des Internets, Uni Paderborn, 2003]

How to show Efficiency?

- **Problem:** Players use discrete increments/decrements when reacting to the feedback. Thus the load oscillates and does not converge.



[C. Schindelhauer, Algorithmische Grundlagen des Internets, Uni Paderborn, 2003]

Efficiency Analysis (1)

- An efficient situation is reached, if the load oscillates within a bounded interval around $E(x)=0$
  - If the load is below the knee ($X(t) < K$), then the overall load has to increase in the next round: $X(t+1) > X(t)$
  - If the knee load is exceeded ($X(t) > K$), then the overall load has to decrease in the next round: $X(t+1) < X(t)$

[C. Schindelhauer, Algorithmische Grundlagen des Internets, Uni Paderborn, 2003]

Efficiency Analysis (2)

- **For higher load X(t) > K:**

$$
\begin{aligned}
X(t+1) < X(t) \quad &\Leftrightarrow \quad \sum_{i=1}^{n} x_i(t+1) < \sum_{i=1}^{n} x_i(t) \\
&\Leftrightarrow \quad \sum_{i=1}^{n} a_D + b_D x_i(t) < X(t) \\
&\Leftrightarrow \quad n\, a_D + b_D X(t) < X(t) \\
&\Leftrightarrow \quad b_D < 1 - \frac{n\, a_D}{X(t)}
\end{aligned}
$$

- $a_D \leq 0 \Rightarrow b_D < 1$
- $a_D > 0 \Rightarrow b_D$ has to be negative - not possible

[C. Schindelhauer, Algorithmische Grundlagen des Internets, Uni Paderborn, 2003]

Efficiency Analysis (3)

- **For lower load X(t) < K:**

$$
\begin{aligned}
X(t+1) > X(t) \quad &\Leftrightarrow \quad \sum_{i=1}^{n} x_i(t+1) > \sum_{i=1}^{n} x_i(t) \\
&\Leftrightarrow \quad \sum_{i=1}^{n} a_I + b_I x_i(t) > X(t) \\
&\Leftrightarrow \quad n\, a_I + b_I X(t) > X(t) \\
&\Leftrightarrow \quad b_D > 1 - \frac{n\, a_I}{X(t)}
\end{aligned}
$$

- $a_I \geq 0 \Rightarrow b_I \geq 1$
- $a_I < 0$: if a=-1 then $b_I > 1 + n/X(t)$, i.e. b_D depends on n (this is not desired)

[C. Schindelhauer, Algorithmische Grundlagen des Internets, Uni Paderborn, 2003]

Fairness Analysis (1)

- Fairness should converge towards 1, i.e. $\lim_{t\to\infty} F(x(t)) = 1$
- Convergence criterion:
  - F(x) is bounded above by 1
  - F(x(t+1)) - F(x(t)) is growing for an appropriate choice of a and b

$$F(x(t)) = \underbrace{\frac{\left(\sum_{i=1}^{n} x_i(t)\right)^2}{n\sum_{i=1}^{n}(x_i(t))^2}}_{\leq 1}$$

$$\begin{aligned}F(x(t+1)) &= \frac{\left(\sum_{i=1}^{n} x_i(t+1)\right)^2}{n\sum_{i=1}^{n}(x_i(t+1))^2}\\ &= \frac{\left(\sum_{i=1}^{n} a+bx_i\right)^2}{n\sum_{i=1}^{n}(a+bx_i)^2}\\ &= \frac{\left(\sum_{i=1}^{n} \frac{a}{b}+x_i\right)^2}{n\sum_{i=1}^{n}(\frac{a}{b}+x_i)^2}\end{aligned}$$

Fairness Analysis (2)

$$F(x(t+1)) - F(x(t)) = \frac{\left(\sum_{i=1}^{n} \frac{a}{b} + x_i\right)^2}{n\sum_{i=1}^{n}(\frac{a}{b} + x_i)^2} - \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n\sum_{i=1}^{n}(x_i)^2} \geq 0$$

common denominator is positive and can be omitted

$$\left(\sum_{i=1}^{n} \frac{a}{b} + x_i\right)^2 \sum_{i=1}^{n}(x_i)^2 - \sum_{i=1}^{n} \left(\frac{a}{b} + x_i\right)^2 X^2 \geq 0$$

$$\left(n\frac{a}{b} + X\right)^2 \sum_{i=1}^{n}(x_i)^2 - \sum_{i=1}^{n} \left(\frac{a}{b} + x_i\right)^2 X^2 \geq 0$$

$$\left(n\frac{a}{b} + X\right)^2 \sum_{i=1}^{n}(x_i)^2 - \sum_{i=1}^{n} \left((\frac{a}{b})^2 + 2\frac{a}{b}x_i + x_i^2\right) X^2 \geq 0$$

expand and remove $X^2 \cdot \sum x_i^2 - \sum x_i^2 \cdot X^2$

$$\left(2\frac{a}{b}X + \underbrace{n(\frac{a}{b})^2}_{\geq 0}\right) \underbrace{\left(n\sum_{i=1}^{n} x_i^2 - X^2\right)}_{\geq 0} \geq 0$$

a/b has to be ≥0

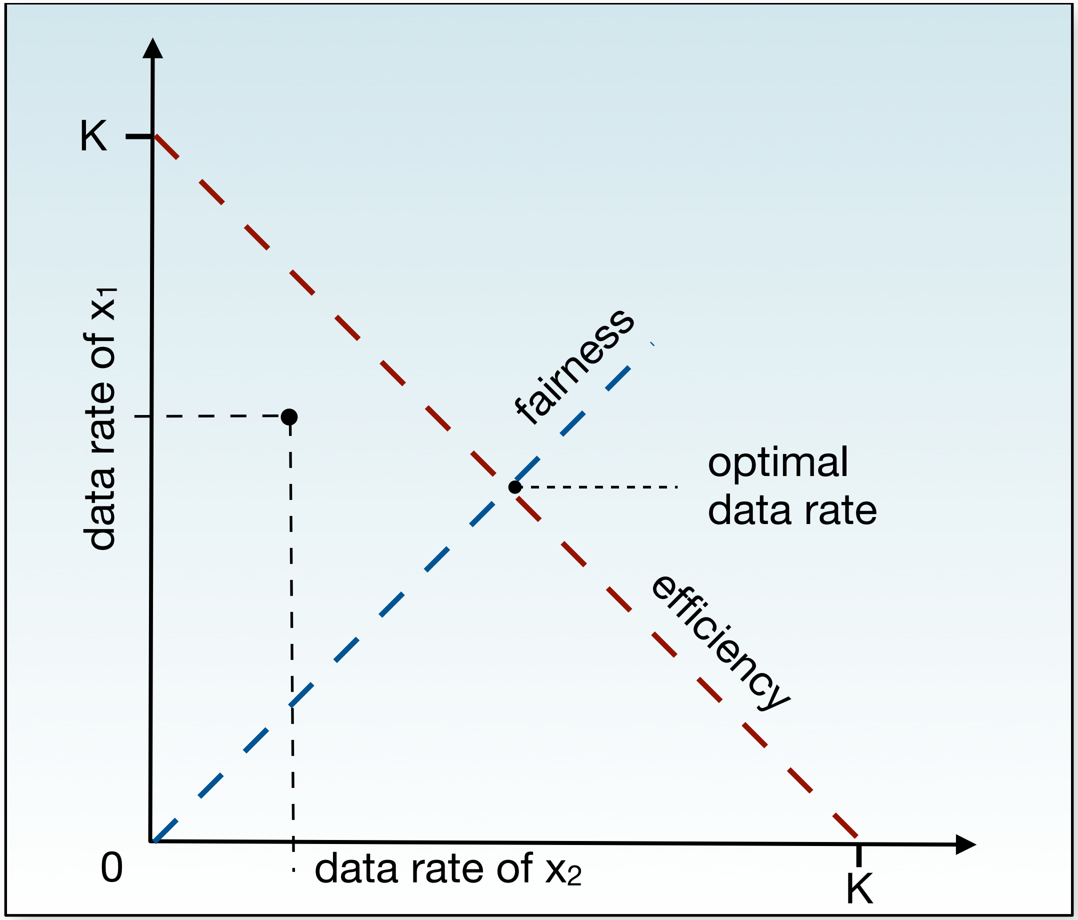
if a/b = 0 then no fairness increase

Parameter selection

- **From efficiency and fairness analysis:**
  - $a_I \geq 0$ and $b_I \geq 1$
  - $a_D \leq 0$ and $a_D \geq 0$, thus $a_D = 0$
  - $0 < b_D < 1$
- $a_D = 0$ means: fairness remains at the same level in the decrease step.
- Fairness can only be reached through the increase step, i.e. $a_I > 0$
- **Summary**: Fairness and efficiency can be reached by an *additive increase* and a *multiplicative decrease*

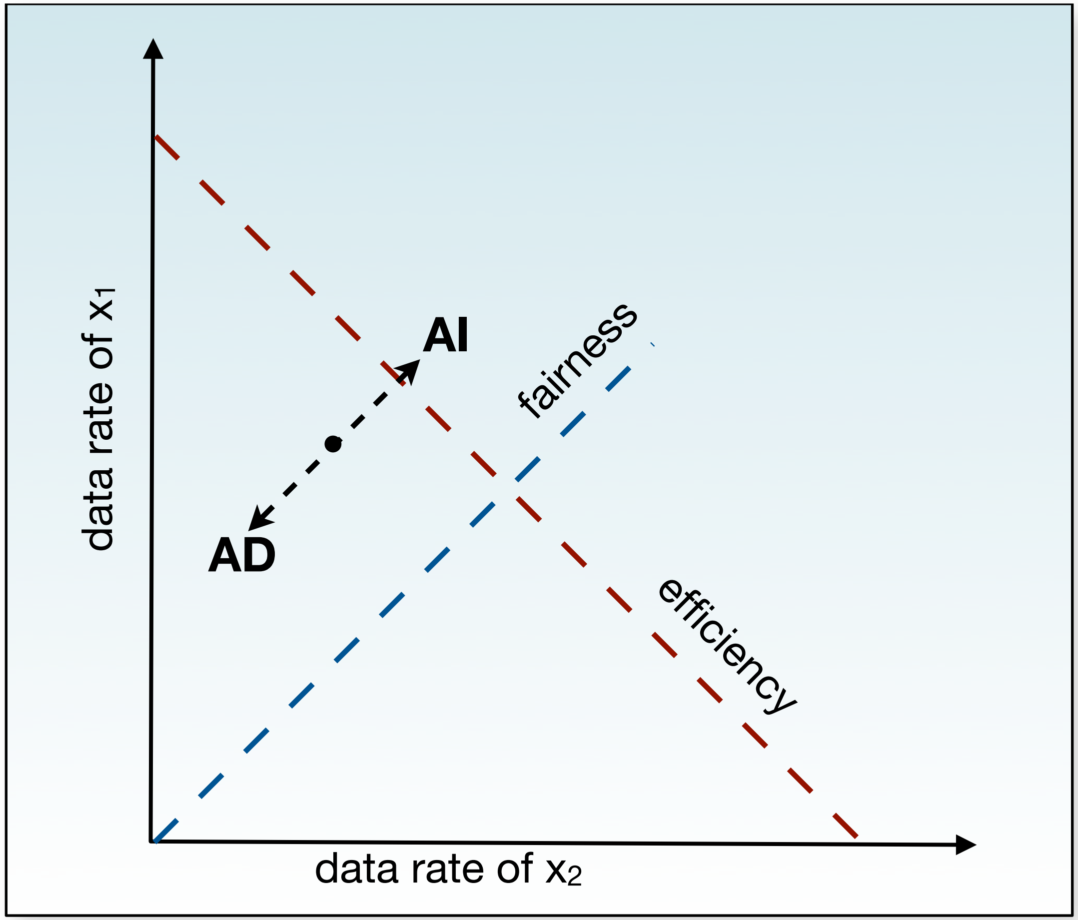
$$f(x, y) = \begin{cases} a_I + x, & \text{if } y(t) = 0 \\ b_D x, & \text{if } y(t) = 1 \end{cases}$$

Vector diagram for 2 participants



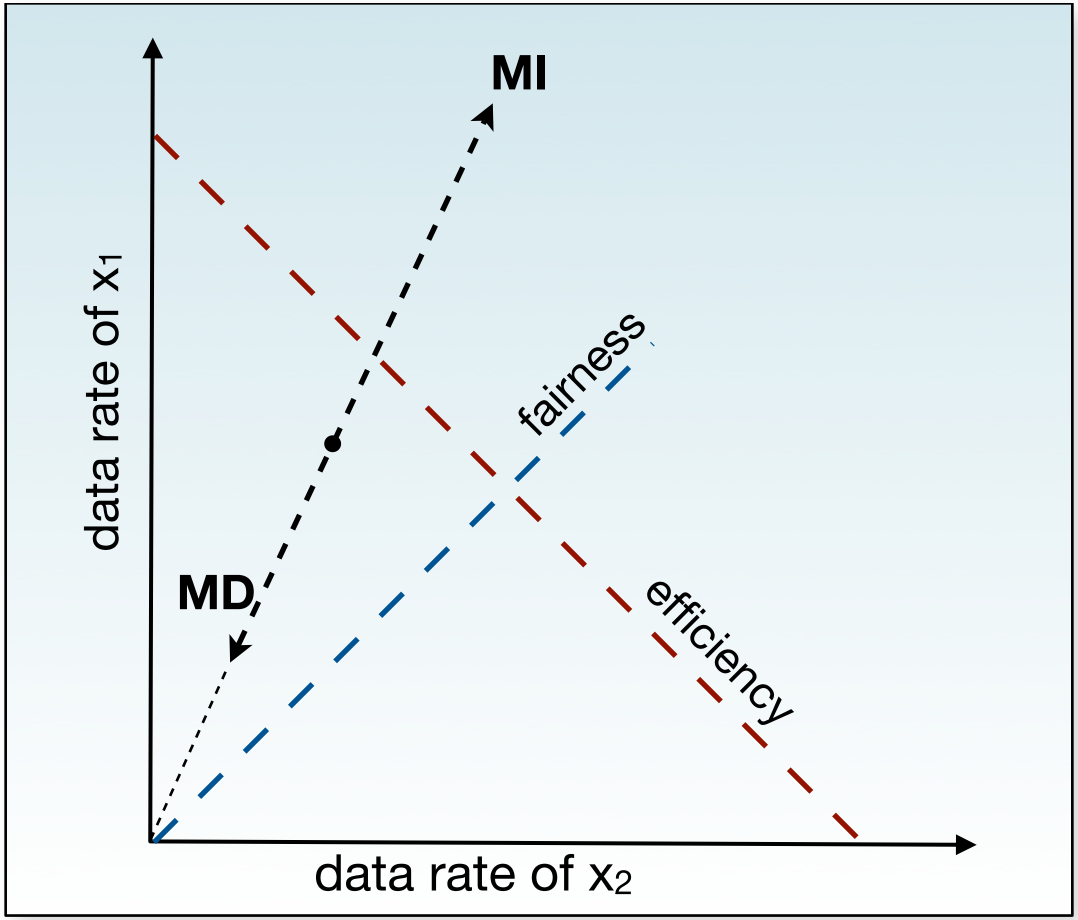
[C. Schindelhauer, Algorithmische Grundlagen des Internets, Uni Paderborn, 2003]

AIAD - Additive Increase/ Additive Decrease



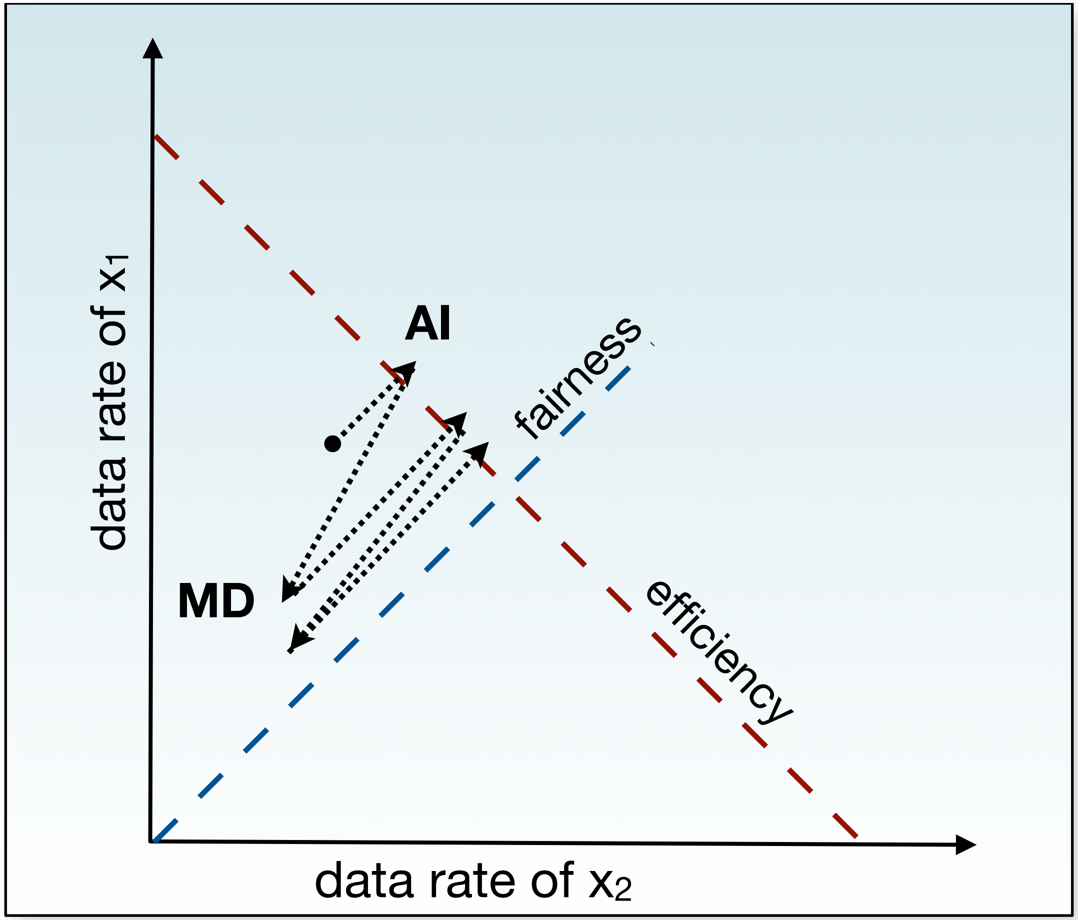
[C. Schindelhauer, Algorithmische Grundlagen des Internets, Uni Paderborn, 2003]

MIMD - Multiplicative Increase/ Multiplicative Decrease



[C. Schindelhauer, Algorithmische Grundlagen des Internets, Uni Paderborn, 2003]

AIMD - Additive Increase/ Multiplicatively Decrease



[C. Schindelhauer, Algorithmische Grundlagen des Internets, Uni Paderborn, 2003]

Conclusion

- Analytical peformance evaluation is based on abstract models
- It requires in-depth knowledge of the system
(can be performed along with experiments/simulations to check whether model abstractions are valid)
- Side-effects should not be neglected due to abstraction