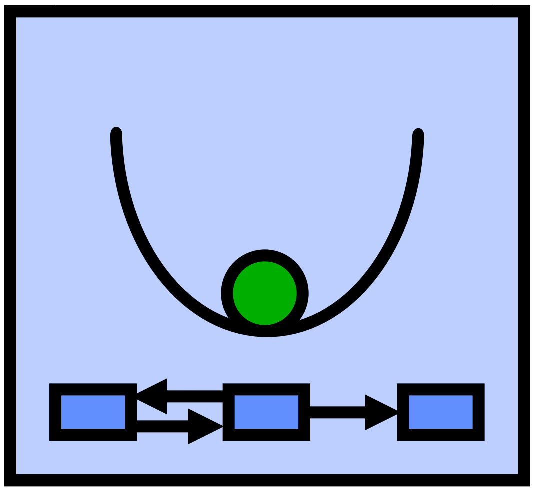
Stability Analysis in Communication Networks

Seminar-Presentation

Presentation 30.06.04

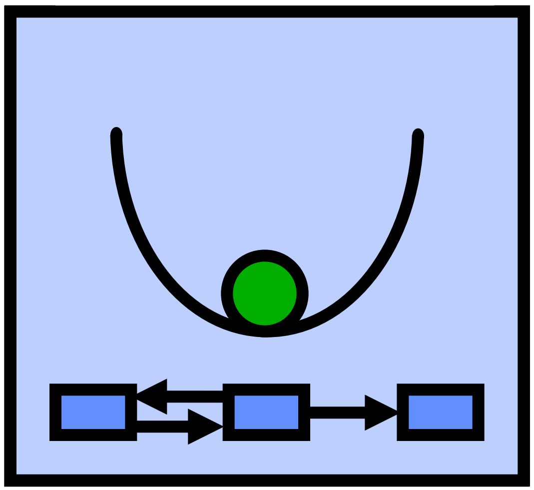


- **Fluid Models**
- **Stability Methods**
- **Conclusion**

Andreas Pillekeit

Institute for Computer Science and
Business Information Systems
University of Duisburg-Essen

Topic Overview



- **Fluid Models**
- **Stability Methods**
- **Conclusion**

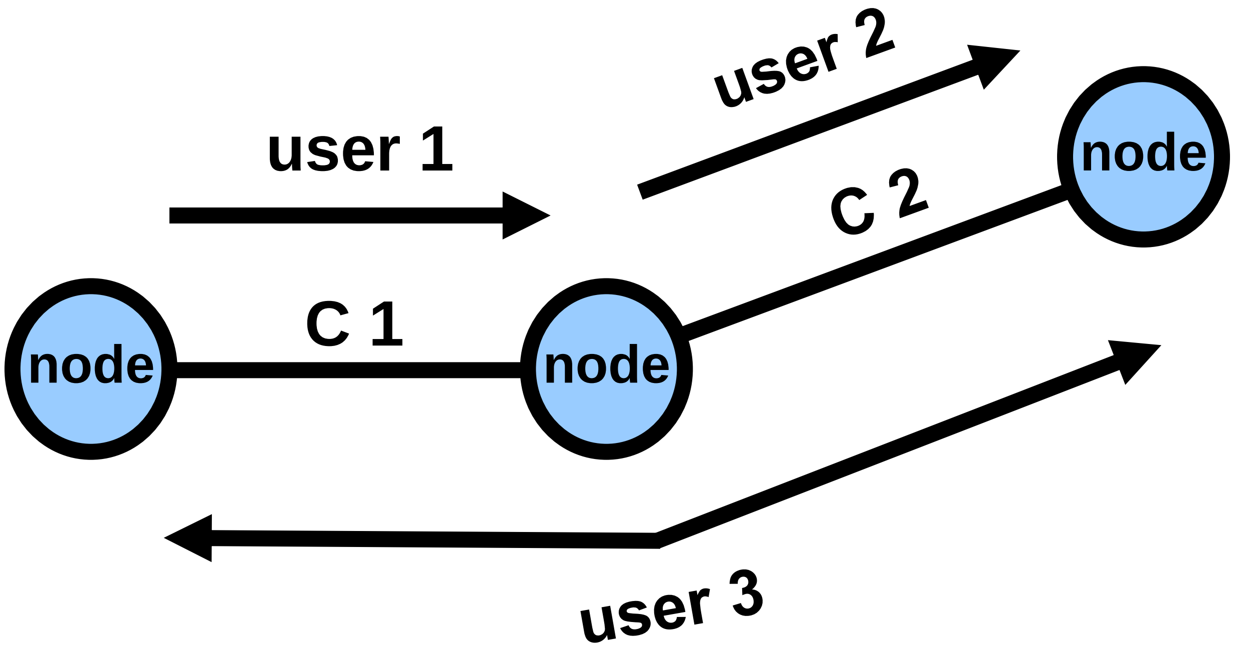
Andreas Pillekeit
Institute for Computer Science and Business Information Systems
University of Duisburg-Essen

Fluid Models – Working groups

- Fluid Models, Game Theory, Randomized Algorithms
  - Tamer Başar, Tansu Alpcan, R. Srikant, Decision and Control Laboratory, University of Illinois, Champaign-Urbana
- Fluid Models, TCP/IP
  - Yong Liu, Francesco Lo Presti, Vishal Misra, Don Towsley and Yu Gu, University of Massachusetts, Computer Science Department, Amherst
    - http://www-net.cs.umass.edu/fluid/

Fluid Models - Introduction 1

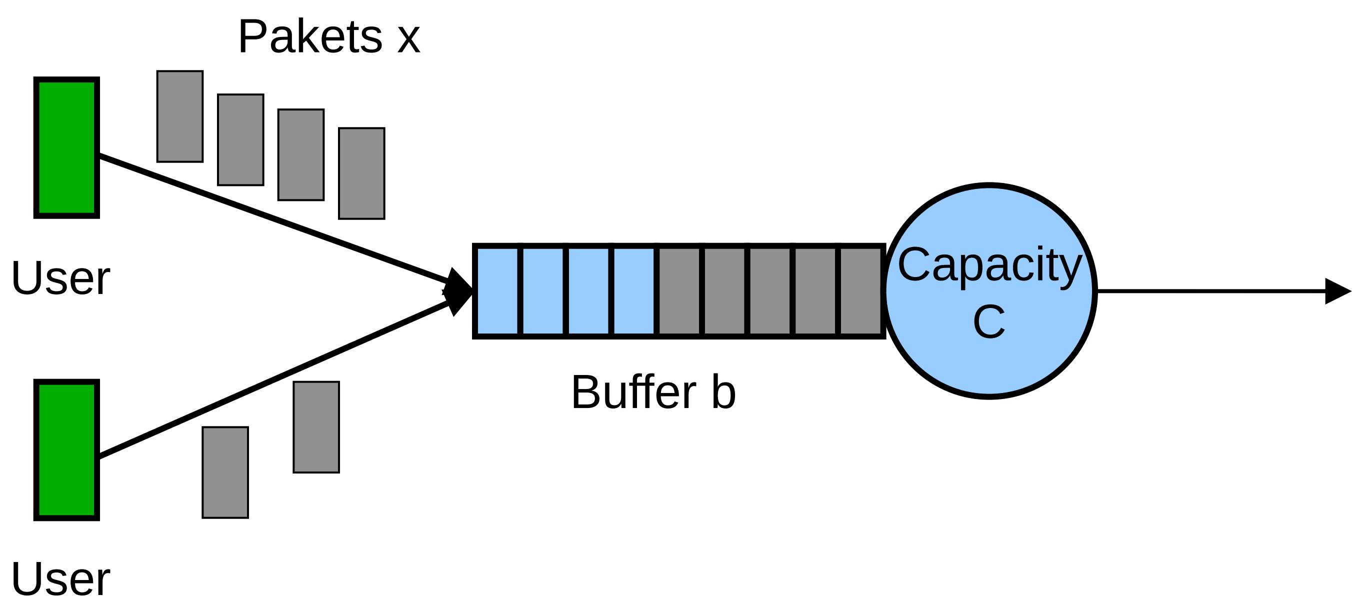
- Arbitrary network structure



Andreas Pillekeit

Institute for Computer Science and Business Information Systems
University of Duisburg-Essen

Fluid Models - Introduction 1

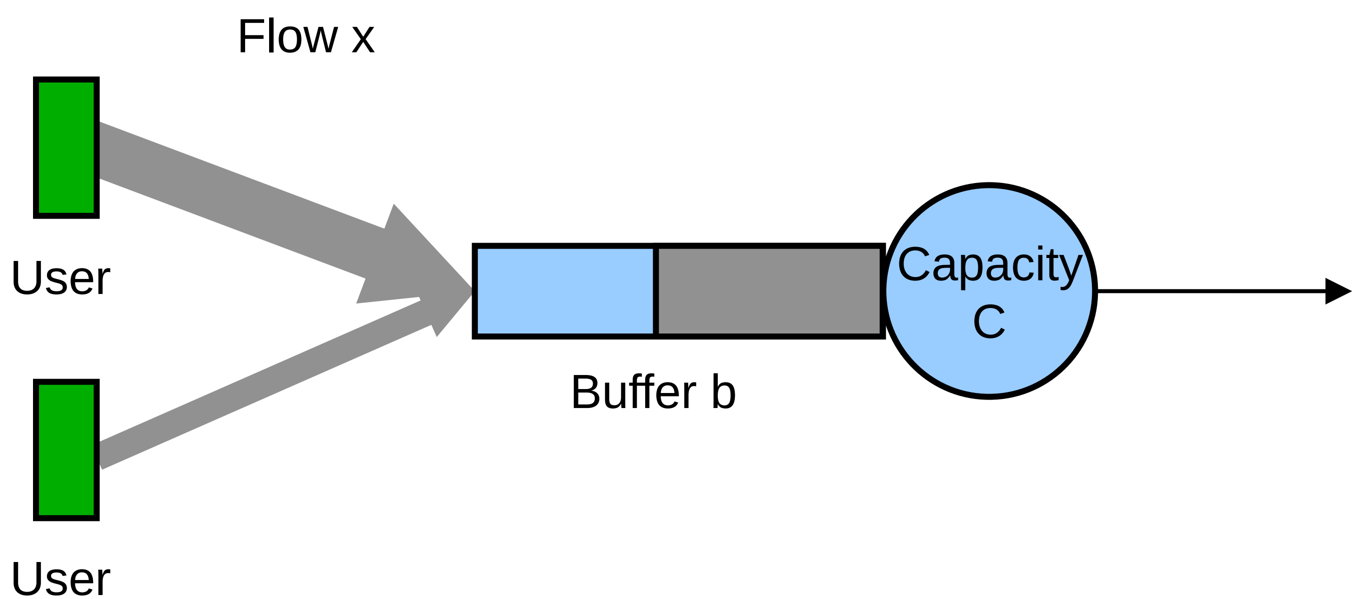


- Data packets and buffer level are described through discrete values

Andreas Pillekeit

Institute for Computer Science and Business Information Systems
University of Duisburg-Essen

Fluid Models - Introduction 2



- Now represented through continuous values
- Real world analogy: Flow of a fluid

Andreas Pillekeit

Institute for Computer Science and Business Information Systems
University of Duisburg-Essen

Fluid Models – System Dynamics 1

$$\dot{b}_l(t) = \begin{cases} \left[\bar{x}_l - C_l\right]^-, & b_l(t) = b_{l,\max} \\ \bar{x}_l - C_l, & 0 < b_l(t) < b_{l,\max} \\ \left[\bar{x}_l - C_l\right]^+, & b_l(t) = 0 \end{cases}$$

- Change of buffer level at node *l*
  = incoming flow – capacity of the outgoing link

Andreas Pillekeit
Institute for Computer Science and Business Information Systems
University of Duisburg-Essen

Fluid Models – System Dynamics 2

$$\dot{d}_l(x,t) = \begin{cases} \left[\frac{1}{C_l} \left(\bar{x}_l - C_l \right) \right]^-, & d_l(t) = d_{l,\max} \\ \frac{1}{C_l} \left(\bar{x}_l - C_l \right), & 0 < d_l(t) < d_{l,\max} \\ \left[\frac{1}{C_l} \left(\bar{x}_l - C_l \right) \right]^+, & d_l(t) = 0 \end{cases}$$

- Change of delay at node *l* (FCFS) = change of buffer level / capacity of the outgoing link

Andreas Pillekeit
Institute for Computer Science and Business Information Systems
University of Duisburg-Essen

Fluid Models – System Dynamics 3

Price → α_i; **Delay over all links used by user *i*** → $D_i(t)$; **Utility** → $U_i(x_i)$

$$J_i(x,t) = \alpha_i D_i(t) x_i - U_i(x_i)$$

$$\dot{x}_i = -\partial J_i(x,t) / \partial x_i$$

- Adaptation of flow rate in accordance with cost function *J*
- Game theory: Unique point where the costs per player are minimized and every change of flow increases the cost of the player (Nash Equilibrium)

Andreas Pillekeit
Institute for Computer Science and Business Information Systems
University of Duisburg-Essen

Fluid Models – System Dynamics 4

- The system dynamics are described through the following differential equations.
  - ignoring the effects of boundaries

$$\dot{x}_i(t) = \frac{dU_i(x_i)}{dx_i} - \alpha_i D_i(x,t)$$

$$\dot{d}_l(t) = \frac{\bar{x}_l}{C_l} - 1$$

Andreas Pillekeit
Institute for Computer Science and Business Information Systems
University of Duisburg-Essen

Fluid Models – TCP System Dynamics 1

- Model of the congestion avoidance part of the TCP slow-start algorithm
  - Bottleneck situation, without timeout mechanism
- TCP increases window size by one every RTT (linear)
- If a congestion occurs the window size will be reduced to half of the original size

$$\dot{W}(t) = \frac{1}{R(t)} - \frac{W(t)}{2} \cdot \underbrace{\frac{W(t-R(t))}{R(t-R(t))} \cdot p(t-R(t))}_{\lambda_C}$$

Fluid Models – TCP System Dynamics 2

- Change of queue length depends on the capacity of the link and the incoming flow

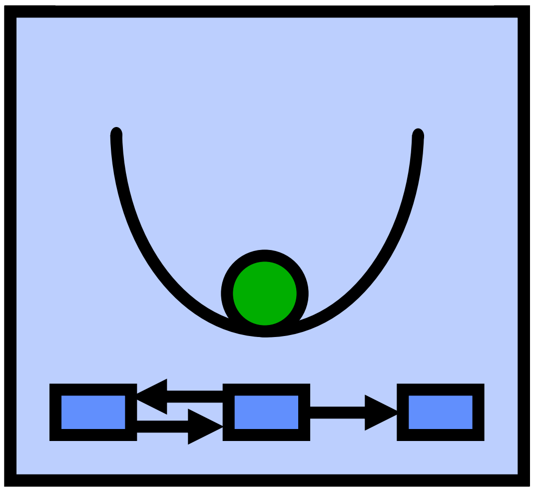
$$\dot{q}(t) = \underbrace{N(t) \cdot \frac{W(t)}{R(t)}} - C$$

Incoming flow = number of connections (N) and sent data per RTT (W/R)

Andreas Pillekeit

Institute for Computer Science and Business Information Systems
University of Duisburg-Essen

Topic Overview



- Fluid Models
- **Stability Methods**
- Conclusion

Andreas Pillekeit

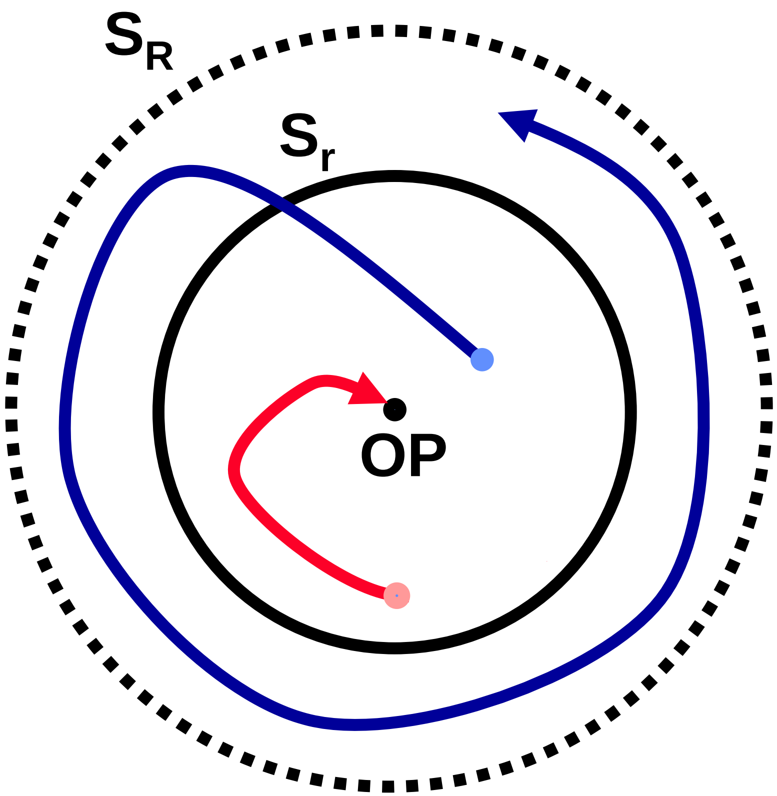
Institute for Computer Science and
Business Information Systems
University of Duisburg-Essen

Stability Methods – Introduction

- Linearization of the nonlinear model around the operating point (OP) to investigate its behavior
  - i.e. Taylor series expansion
  - OP: gradient is zero
- Prediction of stability:
  - energy like Lyapunov function
  - Monte Carlo Simulation
  - Laplace Transformation: open/closed loop investigation

Andreas Pillekeit
Institute for Computer Science and Business Information Systems
University of Duisburg-Essen

Stability Methods – Lyapunov Function 1



- Lyapunov stable: Trajectory of the System never leaves area S_R if it starts within S_r
  - For any given R>0 exist at least one r>0
- Asymptotically stable: If Lyapunov stable and trajectory is heading towards OP for t->∞

Andreas Pillekeit

Institute for Computer Science and Business Information Systems
University of Duisburg-Essen

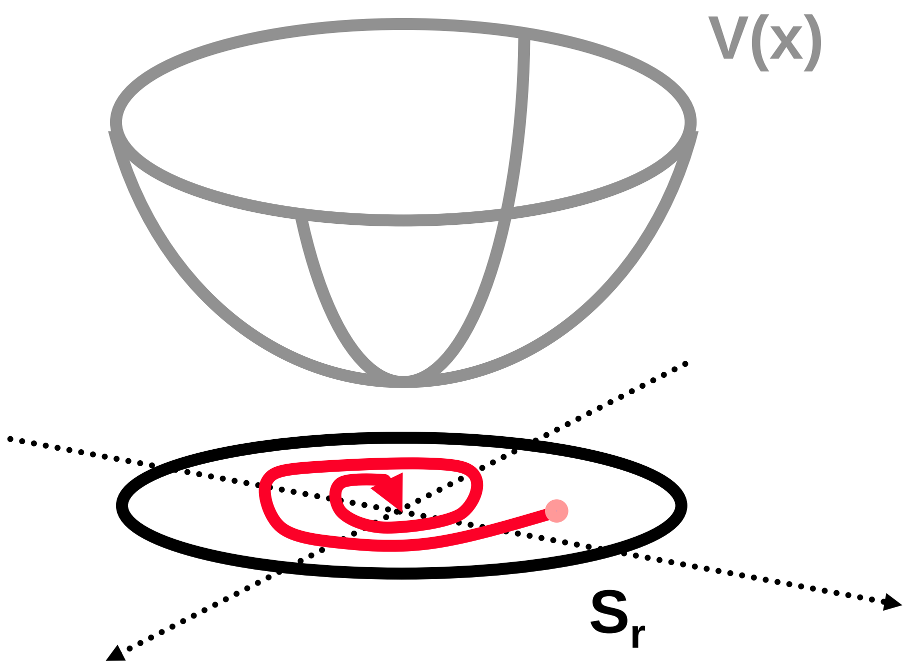
Stability Methods – Lyapunov Function 2

- Analogy: Energy of a system.
  - If a system perpetually loses energy it is asymptotically stable
    - As long as there are no energy inputs
- Generalization: Function V which describes how much energy is in the system
  - Continuously differentiable
  - Positive definite
  - First partial derivative negative (semi)definite
    - Negative definite if asymptotically stable

Andreas Pillekeit

Institute for Computer Science and
Business Information Systems
University of Duisburg-Essen

Stability Methods – Lyapunov Function 3



– Positive definite if V(0)=0 and V(a)>0 for a!=0
  - Energy is in the system if not in OP point
– negative (semi)definite if V'(0)=0 and V'(a)<=0 for all a
  - System loses Energy

Andreas Pillekeit
Institute for Computer Science and Business Information Systems
University of Duisburg-Essen

Stability Methods – Lyapunov Function Example

- The first example system was described through x' and d'
- Amount of data in the flow and the buffer which deviates from the OP(x_L,d_L) is the energy in the system
- V'(x_L,d_L) is negative semidefinite if the utility function U is strictly increasing and concave and the RTT is smaller than a certain computable value

Andreas Pillekeit

Institute for Computer Science and Business Information Systems
University of Duisburg-Essen

Stability Methods – Monte Carlo Simulation 1

- Randomized algorithms are a useful non-analytical tool for investigating stability and robustness of a system
- Monte Carlo methods use random sampling
  - Discover whether the members of a set N have a certain property by randomly choosing a member from that set and testing it
  - Repeating this process obtains a probabilistic estimate on whether the set does have that property

$$\hat{p} = \frac{N_{Stable}}{N_{Total}}$$

Andreas Pillekeit
Institute for Computer Science and Business Information Systems
University of Duisburg-Essen

Stability Methods – Monte Carlo Simulation 2

- Discretized version of the first example system model described in matrix form

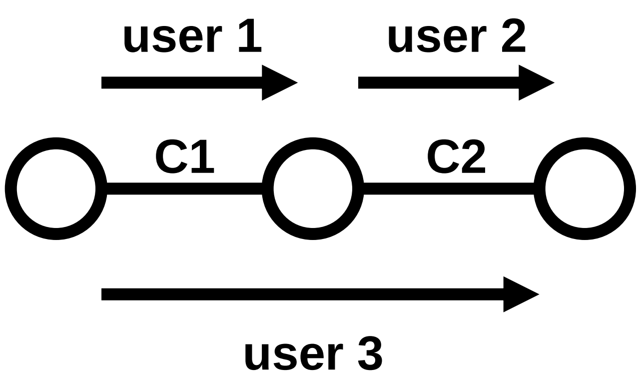
$$\begin{pmatrix} x_L(t+1) \\ d_L(t+1) \end{pmatrix} = G(\alpha, u, C) \begin{pmatrix} x_L(t) \\ d_L(t) \end{pmatrix}$$

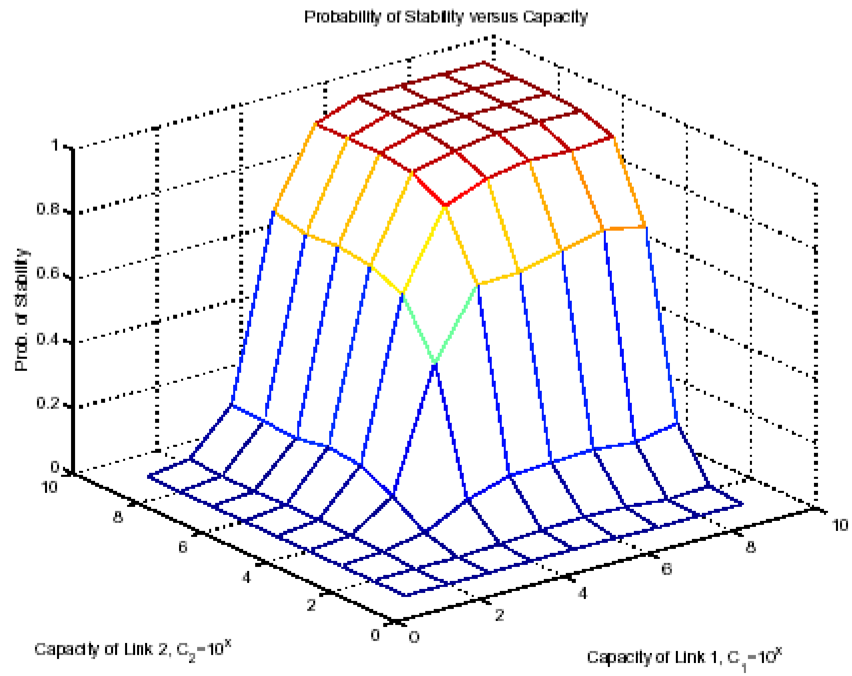
- The discretized system is stable if the absolute values of the eigenvalues of G are less than 1
- Analyze probability of stability for a given parameter space (M-user)

$$p = \left[\alpha_1, \ldots, \alpha_M, u_1, \ldots, u_M, C_1, \ldots, C_L\right]$$

Stability Methods – Monte Carlo Simulation Example

- One Result: Probability of stability increases with increasing capacity of the links
  - 3 Users, 2 Links, Parameter ranges $0<u_i<1$, $0<\alpha_i<1000$, $10^2<C_l<10^6$

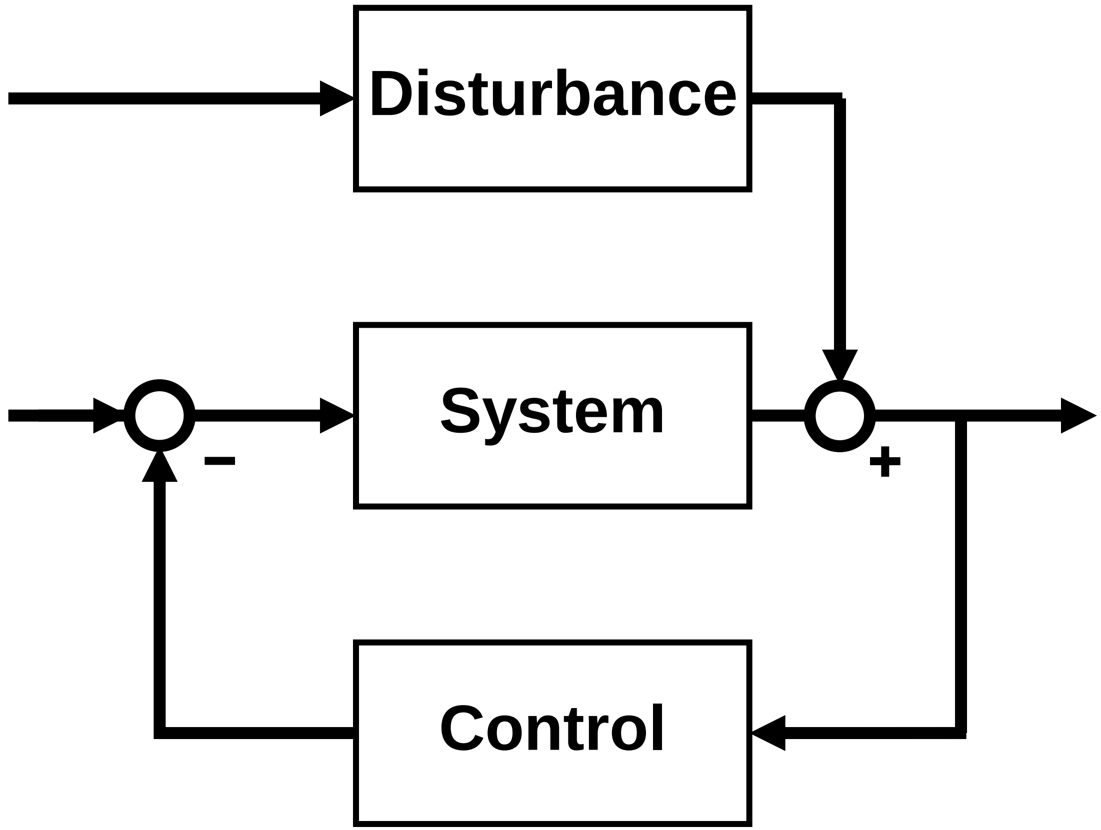




Andreas Pillekeit

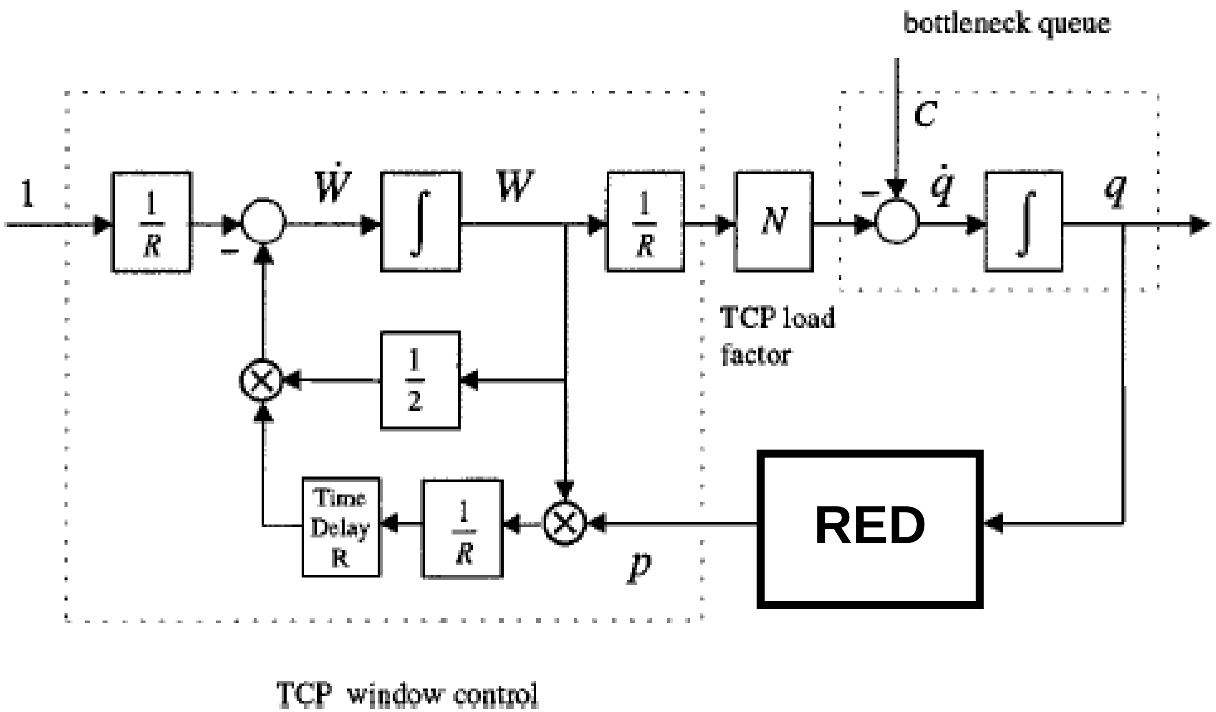
Institute for Computer Science and Business Information Systems
University of Duisburg-Essen

Stability Methods – open/closed loop investigations



Andreas Pillekeit
Institute for Computer Science and Business Information Systems
University of Duisburg-Essen

Stability Methods – TCP System Dynamics



- W= TCP window size
- q= Queue length
- R= Round trip time
- C= Link capacity
- N= Number TCP connections
- p= probability of marked packets
- RED= Random Early Detection

Andreas Pillekeit
Institute for Computer Science and Business Information Systems
University of Duisburg-Essen

Stability Methods – open/closed loop investigations

- Transformation into Laplace space eases solving of integral and differential equations

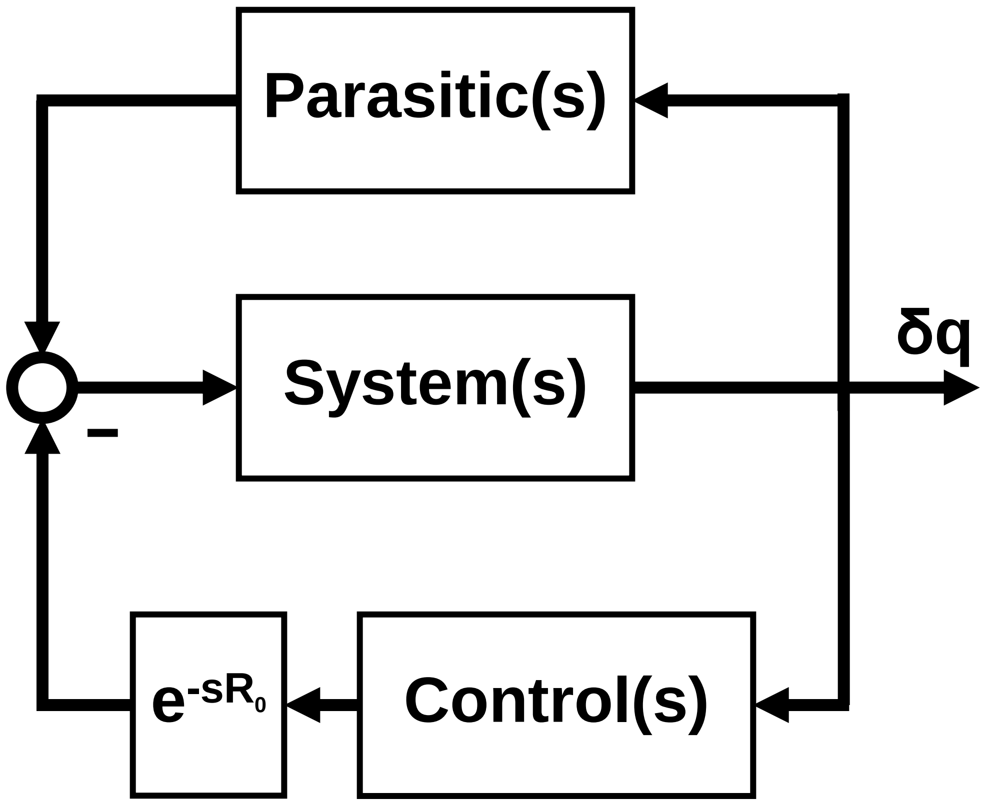
$$L\left[\frac{d^n f(t)}{dt^n}\right] = s^n f(s)$$

$$L\left[\int f(t)dt\right] = \frac{1}{s} f(s)$$

$$s = \sigma + j\omega$$

Stability Methods – open/closed loop investigations

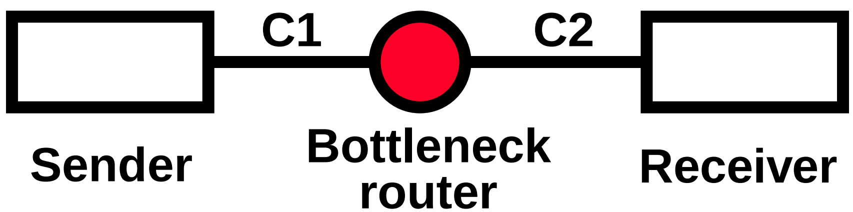
- After linearization and transformation into Laplace space



Andreas Pillekeit
Institute for Computer Science and Business Information Systems
University of Duisburg-Essen

Stability Methods – open/closed loop investigations

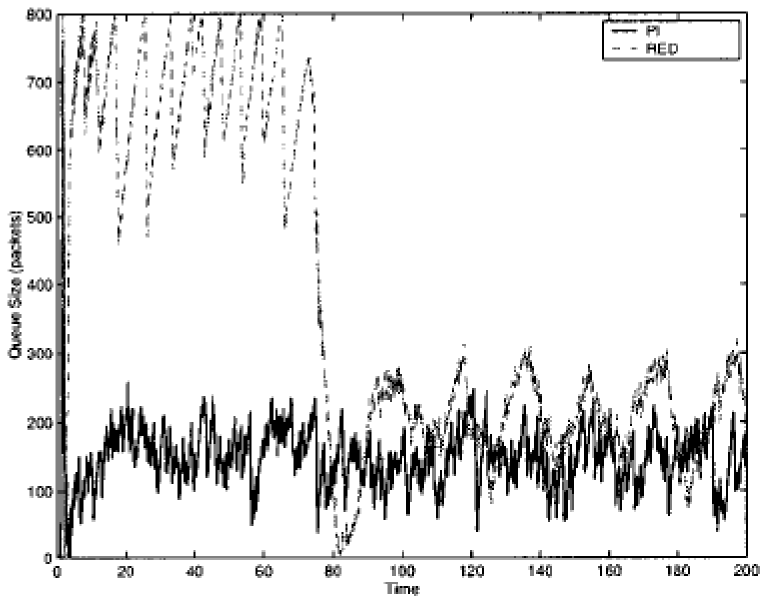
- Instead of using the RED control law a Proportional / Integral PI-Controller was used
- PI-Controller was tuned to the investigated system dynamics of TCP



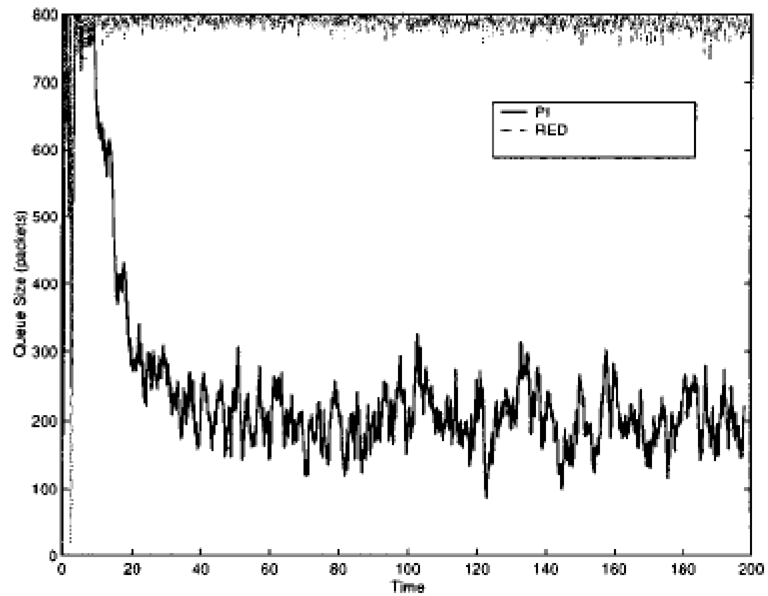
- C1=C2=15Mb/s, average packet size = 500 bytes, propagation delay range uniformly between 160ms and 240ms.
- Light TCP flow: 16 sessions, Heavy TCP flow: 400 sessions

Stability Methods – open/closed loop investigations

- Operating point was chosen as q_0=200 packets
- RED is generally slower than the PI-controller and it is not able to reach the desired OP for the heavy traffic situation



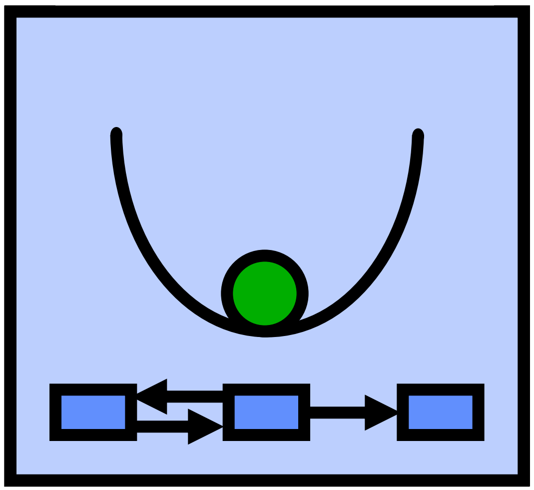
Light TCP flow



Heavy TCP flow

Andreas Pillekeit
Institute for Computer Science and Business Information Systems
University of Duisburg-Essen

Topic Overview



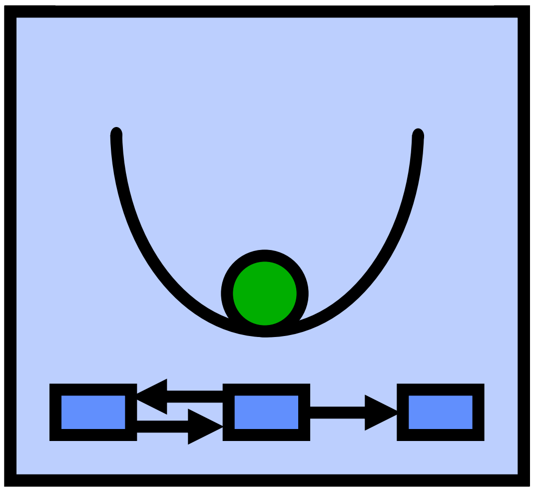
- Fluid Models
- Stability Methods
- Conclusion

Andreas Pillekeit

Institute for Computer Science and Business Information Systems
University of Duisburg-Essen

Conclusion

- Fluid models are capable of simulating the behavior of network algorithms
- Applying control laws known from automation engineering lead to a better understanding and a better performance of network algorithms
- Next step is to create a frame work for investigations of network control problems



Thank you for your attention!

Andreas Pillekeit
Institute for Computer Science and
Business Information Systems
University of Duisburg-Essen