



# Token Bucket Algorithmus

Förg Sebastian, Gritsch Marc, Pühl Gerd, Schoof Martin, Zohner Stephanie



Programmierung verteilter Systeme Lab  
Institut für Informatik  
Universität Augsburg  
Universitätsstraße 14, 86159 Augsburg  
Tel.: (+49) 821/598-2174, Fax: -2175  
URL: <http://www.informatik.uni-augsburg.de/Vs>



# Agenda

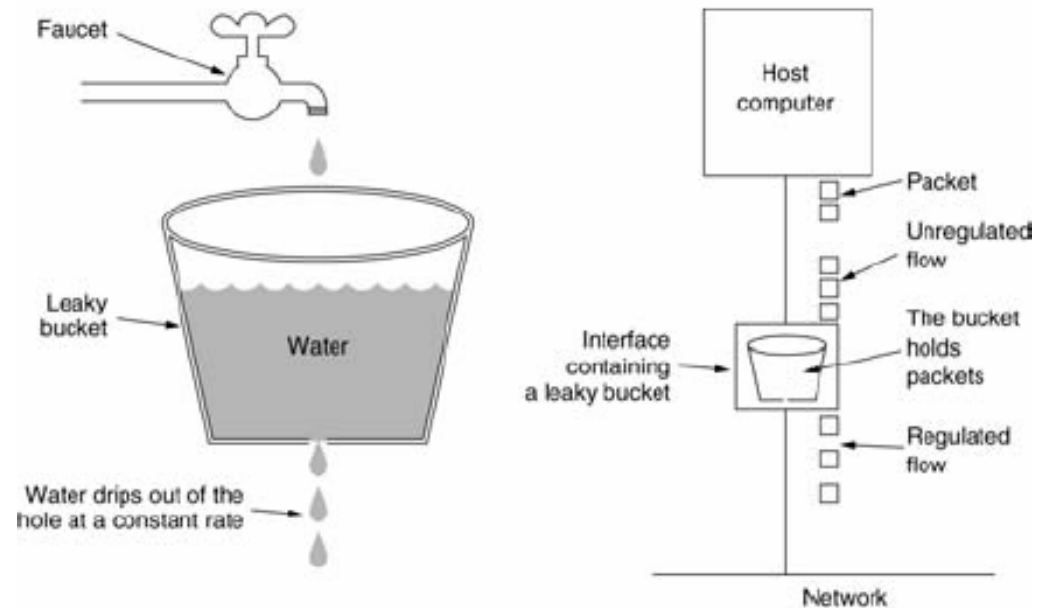


- Motivation
  - ▶ Einordnung in die Vorlesung
  - ▶ Leaky Bucket Algorithmus als Grundprinzip
- Token Bucket Algorithmus
  - ▶ Funktionsweise
  - ▶ Eigenschaften und Abhängigkeiten
- Quality of Service (QoS)
  - ▶ Bessere Vorhersagbarkeit des Netzwerks
  - ▶ QoS in ATM
  - ▶ QoS-Modell der ITU
  - ▶ QoS-Techniken der IETF
- Tool
  - ▶ Vorstellung des Tools
  - ▶ Vorstellung des Quellcodes
  - ▶ Verbesserungen
- Ausblick

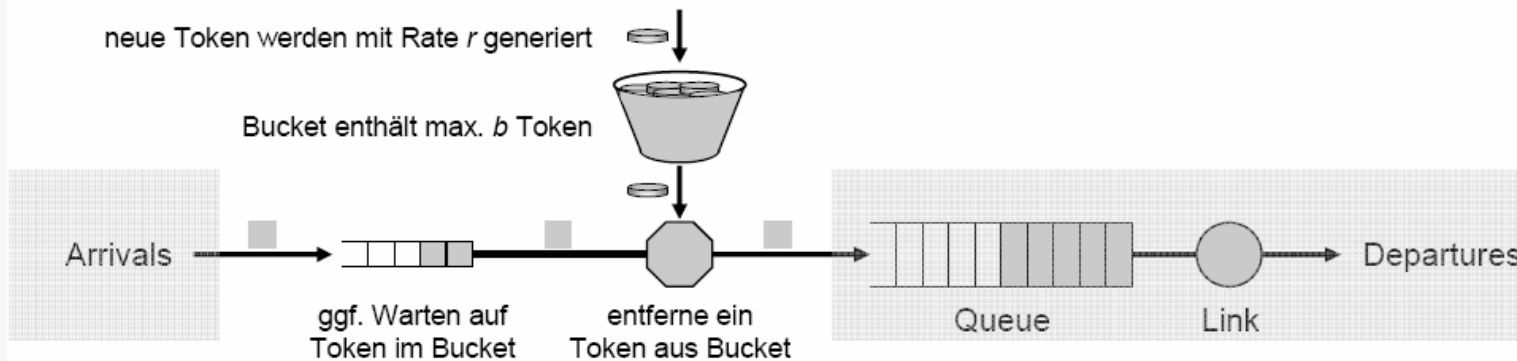


- Einordnung in die Vorlesung
  - ▶ Interprocess Communication
    - ▶▶ Stream oriented Communication
  - ▶ Transfer von Streams (Reihenfolge und Intervalle wichtig)
  
- Probleme
  - ▶ Verteilte Systeme
  - ▶ Beschränkte Bandbreiten
  
- Lösungen
  - ▶ Notwendige Lastverteilung
  - ▶ Überlastung beheben bzw. vermeiden
  - ▶ → Verkehrsformung

## ■ Leaky Bucket Algorithmus



- ▶ Daten werden gleichmäßig übertragen
- ▶ Übertragung wird „geglättet“
- ▶ Keine Traffic-spitzen
- ▶ Bursts werden teilweise verworfen



## ■ Funktionsweise

- ▶ Produktion neuer Token mit bestimmter Rate
- ▶ Bucket kann maximal eine bestimmte Anzahl an Token enthalten
- ▶ Ist die Kapazitätsgrenze des Bucket erreicht, werden neue Token verworfen
- ▶ Für die Übertragung eines Datenpaketes der Länge  $n$  werden  $n$  Token aus dem Bucket entfernt
- ▶ Sind nicht genug Token im Bucket vorhanden, muss das Paket warten

- **Eigenschaften des Algorithmus**
  - ▶ Gleichmäßiger Strom von Datenpaketen entsteht (→ Traffic Shaping)
  - ▶ Durchschnittliche Datenrate kann über die Generierungsrate von neuen Token gesteuert werden
  - ▶ Es gehen keine Datenpakete verloren
  
- **Möglichkeit zur Durchführung von Bursts**
  - ▶ Ansparen von Token, um dann viele Datenpakete auf einmal zu verschicken
  - ▶ Größe des Bursts abhängig von der Kapazitätsgrenze des Buckets
  
- **Abhängigkeiten**
  - ▶ Durchschnittliche Senderate entspricht der Rate  $r$ , mit der neue Token produziert werden
  - ▶ Maximale Senderate wird durch  $r$  und die Bucketgröße  $b$  bestimmt
  - ▶ Burstgröße hängt von  $r$ ,  $b$  und der maximalen Senderate ab



# Quality of Service (QoS)



- Quality of Service (QoS) / Dienstgüte
  - ▶ Verfahren zur Beeinflussung des Datenverkehrs von Netzwerken
  - ▶ vorgegebene Qualitätsparameter
  
- Bessere Vorhersagbarkeit des Netzwerks:
  - ▶ abgestufte Bandbreiten
  - ▶ Verbesserung der Verlusteigenschaften
  - ▶ Vermeidung von Verkehrsstaus
  - ▶ Optimierung des Netzwerkverkehrs
  - ▶ Vergabe von Prioritäten
  
- Drei unterschiedliche Modelle:
  - ▶ QoS in ATM (Asynchronous Transfer Mode)
  - ▶ QoS-Modell der ITU (International Telecommunication Union)
  - ▶ QoS-Techniken der IETF (Internet Engineering Task Force)



## ■ QoS in ATM

- ▶ Dienstgütern sind auf Zellebene definiert
- ▶ Client und Server vereinbaren Dienstklasse und Bandbreite („Signalisierung“)
- ▶ Dienstqualitäten haben vier Eigenschaften
- ▶ vier QoS-Klassen (entsprechend den ATM-Dienstklassen)

## ■ QoS der ITU

- ▶ komplexes, hierarchisches QoS-Modell
- ▶ drei QoS-Klassen

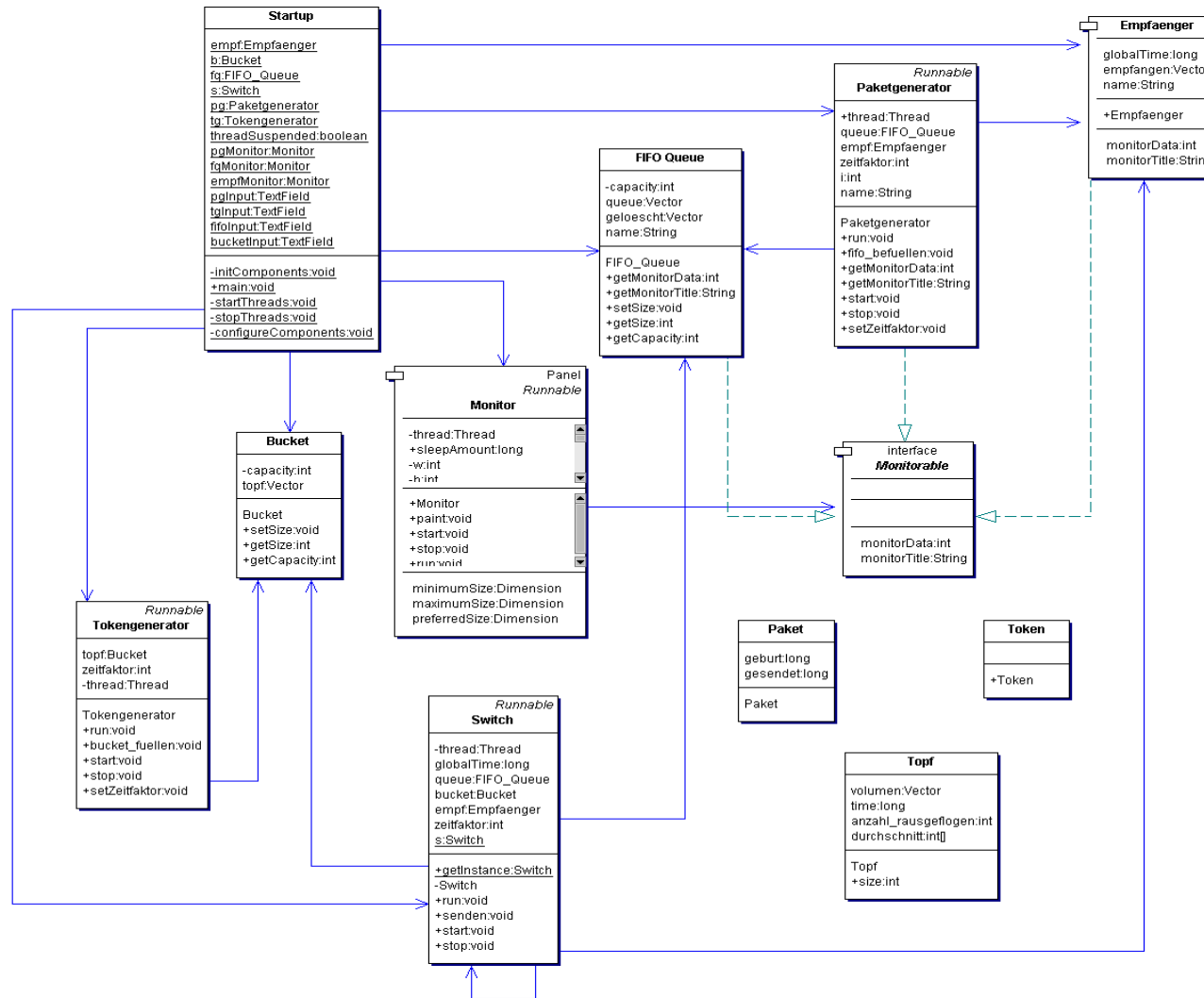


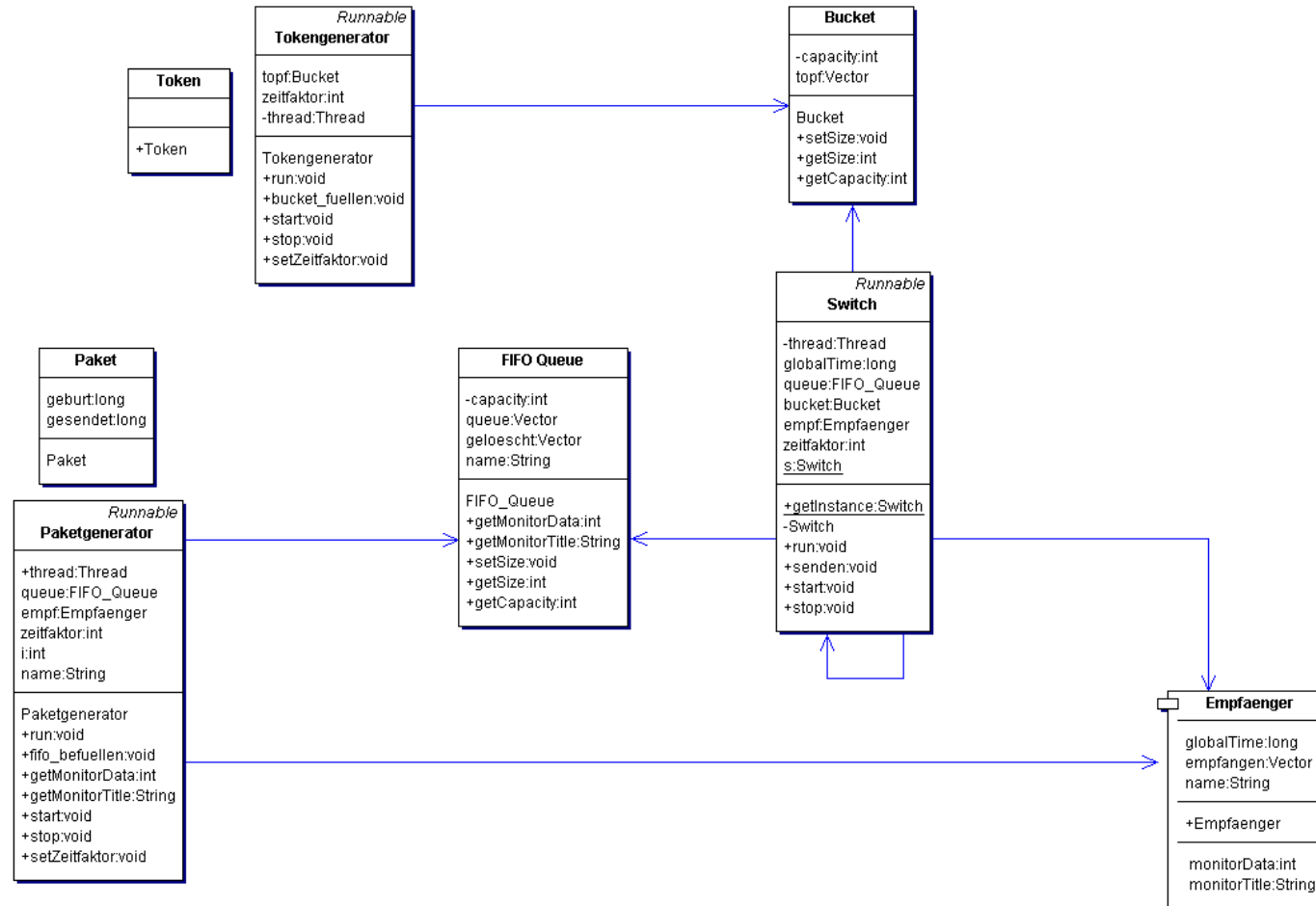


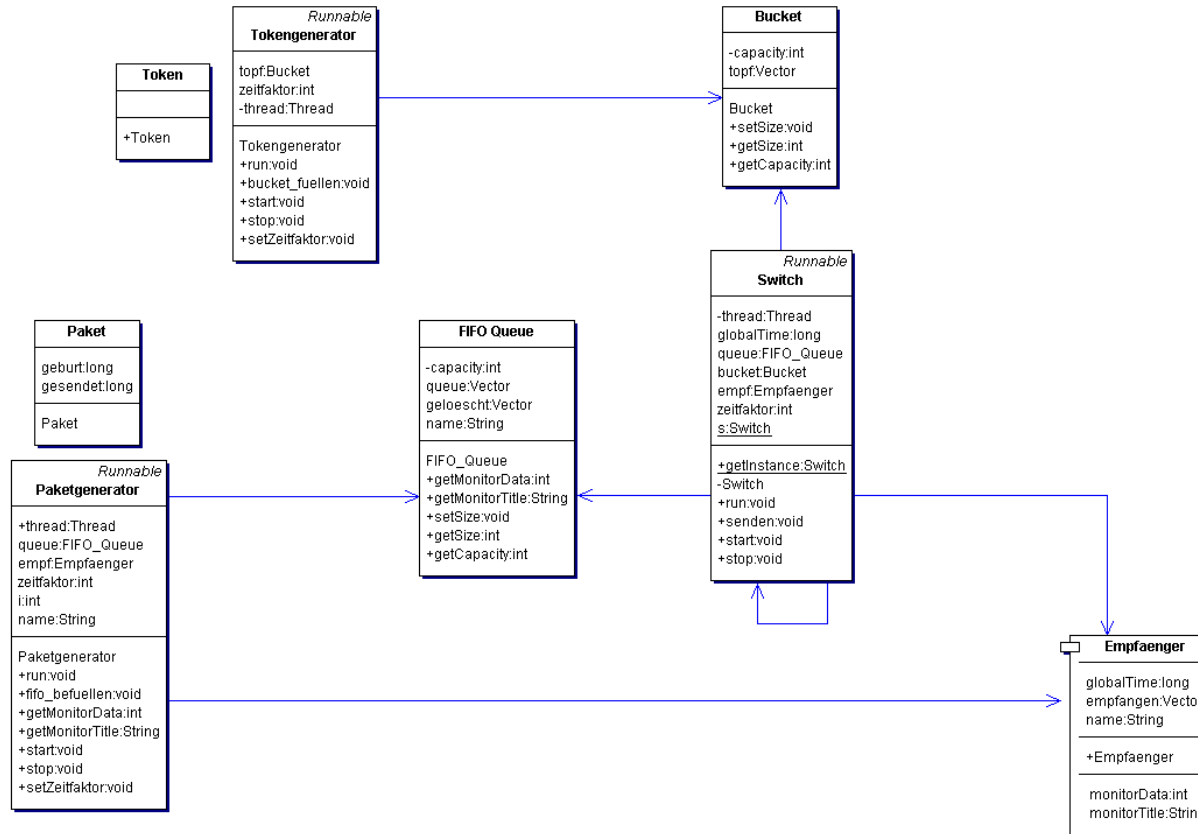
## Quality of Service (QoS)



- QoS-Techniken der IETF
  - ▶ Auch TCP/IP ist mit QoS ausgerüstet
  - ▶ Header von IPv6 enthält QoS-Feld
  - ▶ 16 Stufen (0-15)
  - ▶ Stufen 8-15: Priorität (z.B. Echtzeit)
  
- Drei verschiedene Modelle für das QoS-Verfahren:
  - ▶ Best-Effort-Service (einfachstes Modell, für z.B. E-mail, HTTP, ...)
  - ▶ Integrated Service (Anwendung fordert Dienst mit bestimmter Güte)
  - ▶ Differentiated Service (QoS-Anforderungen werden den einzelnen Datenpaketen entnommen)
  
- Quelle:  
[www.network-projects.telekom.de](http://www.network-projects.telekom.de)



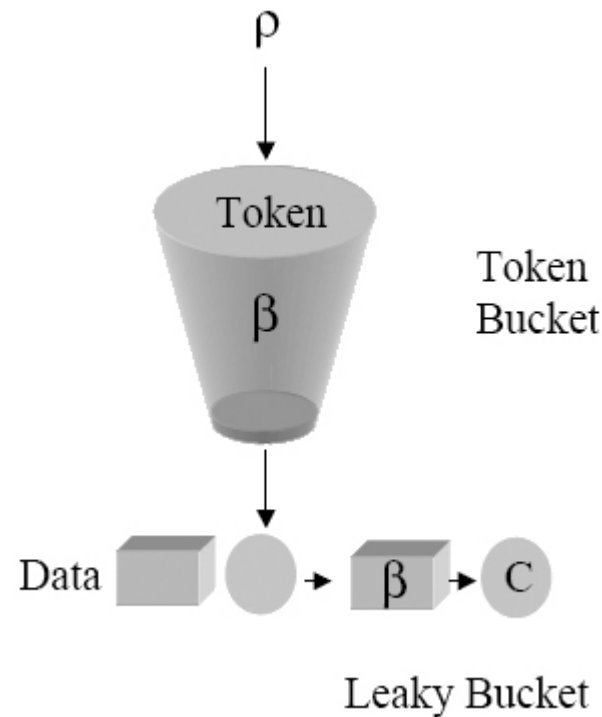




- Problem
  - ▶ Paketgenerierung

## ■ Kombination von Token und Leaky Bucket Algorithmus

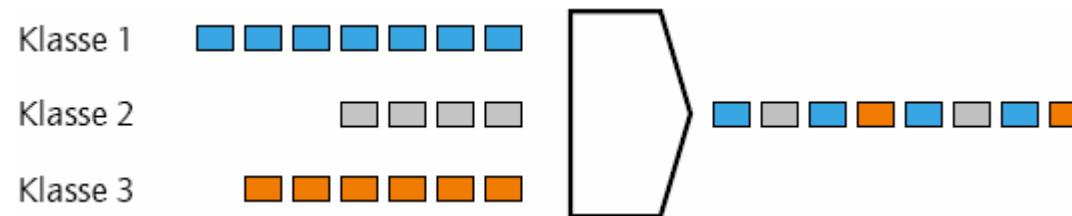
- ▶ Token Bucket erlaubt Bursts die anschließend durch einen Leaky Bucket geglättet werden
- ▶ Beide Buckets haben die gleiche Größe



## ■ Scheduling der Pakete

- ▶ Beispiel: Weighted Fair Queuing
  - ▶ Jeder Flow Typ erhält eigene Warteschlange
  - ▶ Round-Robin mit Bevorzugung (= Gewichtung)

Klasse1 hat Gewicht „2“, Klasse2 und 3 jeweils Gewicht „1“





**Vielen Dank für Ihre  
Aufmerksamkeit!**