Adaptive Load Control Algorithms for 3rd Generation Mobile Networks

Simin Nadjm-Tehrani
Calin Curescu
Linköping University, Sweden

Kayvan Najarian
Teresa A. Dahlberg
University of North Carolina at Charlotte, USA

Tomas Lingvall
Ericsson Radio Systems, Linköping, Sweden

3G Wireless Networks

• Second Generation (2G) Wireless Networks
  • Circuit switched
  • 14 kbps bandwidth
  • Only voice

• Third Generation (3G) onwards
  • Packet switching support
  • 2 Mbps bandwidth
  • Many different multimedia communication and data services

Need for Adaptability

• Nature of the wireless channel
  – Limited spectrum available
  • Overload cannot be solved by over-provisioning
  – Fluctuating nature of the wireless channel
  • Due to fading and interference
  – User mobility
  • Resources available on a cell to cell basis might vary greatly

The Radio Access Network

• UE – user equipment
• Radio base station (RBS)
  • Responsible for communicating with the UE
  • Node B in 3G standard
• Radio Network Controller (RNC)
  • Responsible for managing resources in the radio network, incl. Connection management
  • Mobility management
  • Operations & Maintenance

Problem: Overloads!

• Control the load on a RNC CPU
• Lock of a RNC CPU brings down the respective part of the network
• Watchdog which resets the RNC if the CPU is at 100% utilisation

Goals

• To ensure that control CPUs in RNC nodes are not overloaded: load ≤ U_S
• And also to
  – Minimize the number of rejected tasks
  – Preserve a specified QoS depending on
    • User type
    • Task type
User & Task Priorities

- User types: "Golden" users, emergency calls
- Task types
  - "channel switches" usually more important than "new connections"
  - Direct load – \( U_d \) generated by tasks not under our control

Types of Uncertainty

- Direct tasks - \( U_d \), how much? how often?
- How long does every task take?
  - each task’s utilisation \( c_i \)
- Mix of requested user and task types
- We can measure only the actual utilisation on the processor

Our Approach

- Feedback control (P-controller) for adaptation
  - treat uncertainties in \( c_i \) and \( U_d \)
- Deterministic algorithms to implement the acceptance policies
  - treat uncertainties in mix of tasks and user types

Load Control Schematics

- \( U_{\text{avail}} \) – available load
- \( U_d \) – direct load
- \( U_s \) – set point
- \( U \) – measured load
- \( r_g \) – allocated shares
- \( p_g \) – allocated
- \( a_i \) – accepted

Pool Algorithm (LC-logic)

allocate a certain CPU quota to each type;
for each type loop:
  - if (requested load < allocated load) then requested load granted;
  - unused load is added to a pool;
  - if (requested load > allocated load) then allocated load granted;
the pool is divided among the requests which exceeded their quota
Simulations

- 3G traffic modelled based on Ericsson information from realistic settings
  - The scenarios were run against a simulated model of the machine together with the operating system
  - The traffic models have different mixes of task types for voice, SMS, e-mail, web-browsing

Baseline - Leaky Bucket

- Amount of water to put in the bucket depends on a weighted average of the measured load

Overload Protection

Cumulative overload: sum in time of the load on the processor, which exceeds the set value

Dealing with $U_d$

- Oscillations caused by an oscillative, large $U_d$ (up to 40% of load)
- P controller reacts quicker than the leaky bucket setting

Differentiation between user types

- 2 user types
- 50 / 50 incoming requests
- 80 / 20 user allocated quota
- Channel switches more important than new connections

Summary

- Adaptive admission control algorithms for controlling the CPU load in a RNC
  - Overload protection
  - QoS enforcement mechanism
- Overload management a very common instance of problems dealt within Real-time systems
- Automatic control (feedback-based) techniques a very common way of dealing with adaptation