End-to-End Mechanisms for QoS Support in Wireless Networks

Torsten Braun
joint work with Matthias Scheidegger, Marco Studer, Ruy de Oliveira

Computer Networks and Distributed Systems
Institute of Computer Science and Applied Mathematics
University of Bern, Switzerland

www.iam.unibe.ch/~rvs
COST 279 Mid Term Seminar, January 22, 2004, Rome
Overview

- Motivation
  - IEEE 802.11 Wireless LANs
- End-to-End Probing and Service Selection
  - Experimental Results
- Endpoint Admission Control
  - Simulation Results
  - Virtual Dropping and Out-of-Band Marking
- TCP in Wireless Multi-Hop Networks
  - RTTs in Wireless Multi-Hop Networks
  - Fuzzy Logic for TCP Congestion Control
  - Future Work
- Conclusions
Motivation

- Increasing multimedia application requirements, but:
  - limited resources in wireless networks
  - limited QoS support in wireless LANs / IP networks
- This requires additional mechanisms in end systems such as
  - end-to-end probing / admission control
  - application / end-to-end (e2e) protocol adaptation
IEEE 802.11 Wireless LANs

- **CSMA/CA**
- Acknowledgements and retransmissions (default: 4) on MAC level
  - Link errors can often be repaired but may lead to congestion in retransmitting nodes → prioritization
  - Packet loss due to link errors or due to congestion
- Nodes far from access point can reduce available bandwidth.
E2E Probing and Service Selection

- **Assumptions**
  - 2 services (high / low) available
  - High service costs more than low service.

- **Approach [IPCCC 2002]**
  - Select low service whenever appropriate
  - Switch to high service whenever low service is not sufficient to meet application requirements
    - Examples: bottleneck due to overload or high link error rate
  - Switch back from high to low service as soon as low service is good again
    - This requires parallel monitoring of low service quality (2 % probing traffic)
    - Implementation with RTP/RTCP
  - Oscillations → random probing periods, hysteresis
Experiment

sender

PCM audio

router

load generator

receiver

Background traffic

load

100 %

100 %

t
Results

- Application reacts rather quickly and selects cheapest appropriate service.
- Further experiments with more service classes and additional metrics (jitter, delay)
Endpoint Admission Control (EAC)

- **Idea**
  - Edge devices (e.g., end systems or access routers) measure impact of probing traffic prior to data transmission.
  - Data transmission will only be started if sufficient QoS can be expected, e.g. loss $< \varepsilon$.

- **Classification [Breslau et al., SIGCOMM 2000]**
  - Probing traffic can be transmitted with the same (in-band) or with a different priority / service (lower than data but higher than best-effort, out-of-band)
  - Routers can mark (e.g., explicit congestion notification) or drop probing traffic.
  - 4 combinations
    - in-band dropping
    - out-of-band dropping
    - in-band marking
    - out-of-band marking
  - Options:
    - slow start (several short intervals with increasing probing rate), early reject (several short intervals), simple (one interval)
EAC Results

- Validation of results [Breslau 2000]: ns simulation of 10 Mbps link
- Admission controlled traffic modelled by Poisson arrival process with average inter-arrival time $\gamma = 3.5$ s
- Flows have exponential lifetime with average = 300 s
- Default traffic sources: exponential on/off times (500 ms), burst rate = 256 kbps, packet size = 125 bytes
- Probing time = 5 s, slow start probing with 1 s intervals

![Graph showing loss probability versus utilization with different values of $\epsilon$.](image)
Virtual Dropping

- Marking: Simulation of a virtual queue with 90% bandwidth and marking those packets that would have been dropped by the virtual queue.
- Idea: do not mark but drop those packets → virtual dropping
  - This can only be applied to out-of-band marking. Otherwise regular data packets would be dropped.
- [Breslau et al. 2000] claims without evaluation that „one could easily achieve exactly the same results doing out-of-band virtual dropping instead of out-of-band marking“
Virtual Dropping / Out-of-Band Marking

- **Exponential (on: 125 ms), burst rate 1024 kbps**

- **Pareto on/off (500 ms), burst rate 256 kbps**

- **Star Wars trace**
Virtual Dropping / Out-of-Band Marking

- Out-of-band marking / virtual dropping allow to detect service unavailability earlier.
- Both behave similar but NOT exactly the same:
  Higher utilization and loss probability for virtual dropping than for out-of-band marking due to several reasons:
  - Virtual dropping reduces congestion level slightly.
  - Higher probability that virtual dropping might not detect losses at the end of the probing phase.
- Differences are smaller for periods with rather high overload.
TCP in Wireless Multi-Hop Networks

- Bad TCP performance because congestion control window is decreased for each lost packet even if caused by wireless link errors or by mobility
- TCP should be able to distinguish packet loss caused by network overload and caused by other reasons.
TCP in Wireless Multi-Hop Networks

- Most approaches are based on special functions in the network or TCP extensions:
  - TCP-Feedback / Explicit Link Failure Notification: nodes notify sender about packet loss reason.
  - ATCP distinguishes loss reasons based on ICMP error messages and explicit congestion notifications
  - TCP DOOR receiver reports to signal out of order events to the sender in order to avoid slow start.

- Problems
  - Mechanisms require new functionality in network nodes or TCP protocol changes
  - Security problem: Can a TCP sender trust each intermediate node?
RTTs in Wireless Multi-Hop Networks

- **Goal**
  - Design a reliable congestion control algorithm without special network feedback or protocol extensions

- **Approach**
  - Observe round trip times (RTTs) to decide whether a packet has been lost due to congestion or due to link errors

- **Problem**
  - RTTs might have a very high variance in wireless networks.
    - **Reasons**
      - MAC level retransmissions
      - Mobility of nodes
      - Changing routes
      - Congestion
      - ...
Fuzzy Logic for TCP Congestion Control

- **Fuzzification**
  - Mapping of (discrete) input values to membership functions with smooth transition from 0 to 1.
  - Determination of RTT mean and delay variance parameters \((t_0, t_1, T_{max})\)

- **Inference**
  - Application of (here: 9) predefined fuzzy rules to mapped inputs
  - Min-max inference method

- **Defuzzification**
  - Results of rules are accumulated to discrete output value
  - Gravity-of-mass method for calculating final result

### Fuzzification

- **Mapping of (discrete) input values to membership functions with smooth transition from 0 to 1.**
- **Determination of RTT mean and delay variance parameters \((t_0, t_1, T_{max})\).**

### Inference

- **Application of (here: 9) predefined fuzzy rules to mapped inputs.**
- **Min-max inference method.**

### Defuzzification

- **Results of rules are accumulated to discrete output value.**
- **Gravity-of-mass method for calculating final result.**

---

**Membership**

- **Mean Delay**
- **Delay variance**
  - **Small**
    - Bit Error
    - Congestion
  - **Medium**
    - Bit Error
    - Uncertain
    - Congestion
  - **Large**
    - Bit Error
    - Bit Error
    - Congestion

---

**Graphs**

- Graph showing membership functions for different states:
  - Bit Error
  - Uncertain
  - Congestion

- Graph showing mean delay with parameters \((t_0, t_1, T_{max})\).
Accuracy of Status Detection

- Simulation experiments with ns-2, stationary 3-hop-scenario
- Fuzzy logic engine collects RTT samples and decides whether link is
  - bit erroneous
  - congested
  - both
- Accuracy increases with the number of samples considered, in particular for bit errors. [NEW2AN 2004]
Detection of Congestion

- Simulation experiment:
  - Low packet error rate: 0 - 5 %
  - Generation of congestion
- Fuzzy engine detects congestion faster than TCP. [WCNC 2004]
TCP Performance

- Experiment: 10% packet error rate, no congestion
- Increased TCP performance, because congestion control does not become active for link errors

[WCNC 2004]
Future Work

- Automatic parameter determination
- Mobility
- More simulations with various scenarios
- Additional metrics
- Improved algorithms
Conclusions

- End-end mechanisms are often necessary to adapt applications and protocols in dynamic network environments.

- Smart algorithms might help to improve application performance.