CS551 Integrated Services Packet Networks

[Clark92a]

Bill Cheng

http://merlot.usc.edu/cs551-f12

Copyright © William C. Cheng

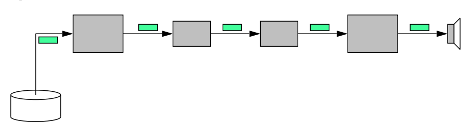
Key Ideas

- Architecture: should allow traffic guarantees
  - guaranteed
  - predicted
  - best effort
  - motivate admission control
- Mechanisms:
  - AQM strategy: FIFO+
  - service interface: token bucket defining rate & burstiness

Copyright © William C. Cheng

Motivation

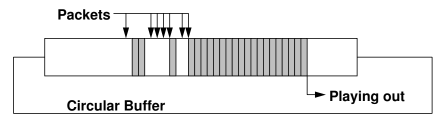
- Some applications require minimum level of network performance
- Some less elastic applications are not able to adapt to changes in bandwidth and delay
  - bandwidth below which video and audio are not intelligible
  - internet telephones, teleconferencing with high delay (200 - 300ms) impair human interaction
- The problem



Copyright © William C. Cheng

A Class of Real-time Applications

- Playback applications
  - set a playback point in the future
  - buffer packets until playback point



- Features that you can leverage
  - early packet arrival ok
  - performance improves with lower delay
  - need absolute or statistical bound on delay
  - tolerate some loss

Copyright © William C. Cheng

Rigid vs. Adaptive Applications

- Two classes of playback applications
  - Rigid/adaptive
  - Tolerant/intolerant
    - the distinction here is whether the application would tolerate interruptions
- Rigid applications
  - Set fixed playback point (*a priori* bound)
- Adaptive applications
  - Adapt playback point (*de facto* bound)
  - A priori bound > *de facto* bound

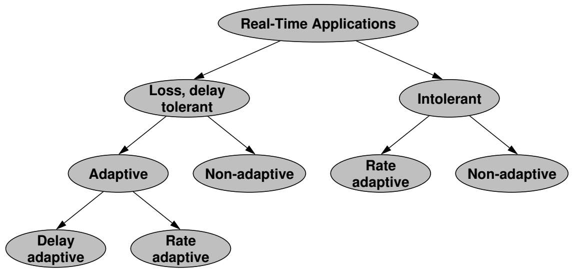
Copyright © William C. Cheng

Adaptive Applications

- Gamble that network conditions will be the same now as in the past
- Are prepared to deal with errors in their estimate
- Will in general have an earlier playback point than rigid applications
- Experience has shown that they can be built (e.g., vat, various adaptive video apps)

Copyright © William C. Cheng

Real-time Applications



Copyright © William C. Cheng

Architectural Components

- Commitments made by network
  - type of service the network provides
- Service interface
  - characterization of source traffic
  - characterization of QoS network will deliver
- Packet scheduling
  - algorithms, information in headers
- Admission control
  - policing

Copyright © William C. Cheng

Types of Network Service Commitments

- Guaranteed service
  - For intolerant and rigid applications
- Predicted service
  - For tolerant and adaptive applications
    - Applications gamble, why not the network?
  - Two components:
    - If conditions do not change, commit to current service
    - If conditions change, take steps to deliver consistent performance (help apps set playback point by minimizing post facto delay bounds)

Copyright © William C. Cheng

Service Interface: Flowspecs

- Tspec: describes the flow’s traffic characteristics
- Rspec: describes the service requested from the network

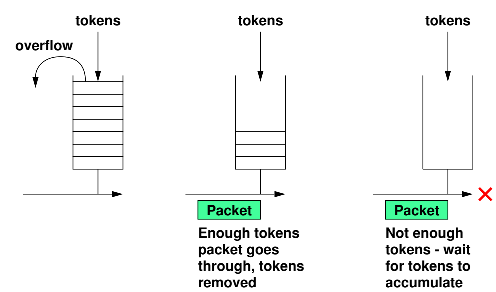
Copyright © William C. Cheng

Token Bucket Filter

- Described by 2 parameters:
  - token rate r: rate of tokens placed in the bucket
  - bucket depth B: capacity of the bucket
- Operation:
  - tokens are placed in bucket at rate r
  - if bucket fills, tokens are discarded
  - sending a packet of size P uses P tokens
  - if bucket has P tokens, packet sent at max rate, else must wait for tokens to accumulate

Copyright © William C. Cheng

Token Bucket Operation



Copyright © William C. Cheng

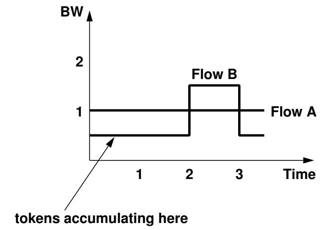
Token Bucket Characteristics

- In the long run, rate is limited to r
- In the short run, a burst of size B can be sent
- Amount of traffic entering at interval T is bounded by:
  - traffic = B + r $\times$ T
- Information useful to admission algorithm

Copyright © William C. Cheng

Token Bucket Specs

- Example:
  - Flow A: r = 1 MBps, B=1 byte
  - Flow B: r = 1 MBps, B=1MB



Copyright © William C. Cheng

Possible Token Bucket Uses

- *Shaping, policing, marking*
  - delay pkts from entering net (shaping)
  - drop pkts that arrive without tokens (policing)
  - let all pkts pass through, mark ones without tokens
    - network drops pkts without tokens in time of congestion

Copyright © William C. Cheng

Guarantee Proven by Parekh

- Suppose a flow
  - gets a rate r at every router in network
  - and all routers in network do WFQ
  - ... and the corresponding token bucket burst size is b
- Then, in any arbitrary topology
  - Cumulative queuing delay Di suffered by flow i has upper bound b/r
  - even if the switch is shared with unshaped flows
- This result holds for a fluid flow approximation
  - Additional terms to the delay bound with a packet approximation
- Intuition:
  - Imagine flow i shaped with token bucket,
  - ... then all delay is incurred at entrance to network

Copyright © William C. Cheng

Scheduling Guaranteed Traffic

- Use token bucket filter to characterize traffic
- Use WFQ at the routers
- Parekh's bound for worst case delay
- Use token bucket filter to characterize traffic
  - Delays can be high unless one reserves a rate r which is higher than the average rate
  - Network can then be significantly underutilized

Copyright © William C. Cheng

Predicted Service

- WFQ not suitable
  - Provides isolation, but the delay is not shared
  - ... and can self-impose jitter in post facto delay
  - FIFO with multiple priority levels might work
    - but jitter can increase in a multi-hop case
- So, use FIFO+ for multi-hop sharing
  - At each hop: measure average delay for class at that router
  - For each packet: compute difference of average delay and delay of that packet in queue
  - Add/subtract difference in packet header

Copyright © William C. Cheng

FIFO+ And Error Diffusion

- FIFO+ has characteristics similar to *error diffusion* in computer graphics
- Original pixel value is an intensity value between 0 (black) and 1 (white)
- Represent the picture in pure black and white
  - *thresholding* -- e.g., replace value by 1 if intensity ≥ 0.5 and replace value by 0 if intensity < 0.5
  - *error diffusion* -- start with thresholding, carry error into the next pixel

original

thresholding

error diffusion

Copyright © William C. Cheng

Predicted Service: FIFO+ Simulation

- Simulation shows:
  - slight increase in delay and jitter for short paths
  - slight decrease in mean delay
  - significant decrease in jitter
- However, more complex queue management

Copyright © William C. Cheng

Unified Scheduling

- Assume 3 types of traffic: guaranteed, predictive, best-effort
- Scheduling: use WFQ in routers
  - each guaranteed flow gets its own queue
  - other traffic aggregates in separate queue
    - predictive traffic classes:
      - several classes separated by order of magnitude delay (sum of delays at each hop)
      - strict priority with FIFO+
    - best effort traffic gets lowest priority

Copyright © William C. Cheng

Service Interface

- Guaranteed traffic
  - specifies rate (but not bucket size, because routers use WFQ in the network and every guaranteed flow gets its own queue)
  - if delay not good, ask for higher rate
- Predicted traffic
  - specifies (r, b)
  - selects delay, loss, network assigns priority
  - policing at edges to drop or tag packets

Copyright © William C. Cheng

Admission Control

- Predicted and guaranteed traffic can overload the network
  - why is this bad? we will fail to satisfy guarantees if we overload the net
  - best-effort not an issue; no guarantees ⇒ ends will back off
- Admission control not addressed in this paper
  - and really hard (who knows what they want?)

Copyright © William C. Cheng

But...

- Do we really need Integrated Services?
  - Do we need to change the network service model?
  - Or, do we just let applications adapt, and engineer the network for enough bandwidth?
- How do we even study this question?

Copyright © William C. Cheng

State of Integrated Services

- Lots of work in the area (e.g., ATM, RSVP)
- We understand many of the problems
  - But no commercial interest in the technology
  - Too complex?
    - Can we build these schedulers in hardware?
    - Need per-flow state for scheduling
  - or is overprovisioning easy
- Can we do something simpler?

Copyright © William C. Cheng