PAC: Perceptive Admission Control for Mobile Wireless Networks

Ian D. Chakeres
Dept. of Electrical and Computer Eng.
Elizabeth M. Belding-Royer
Dept. of Computer Science
University of California, Santa Barbara

Motivation

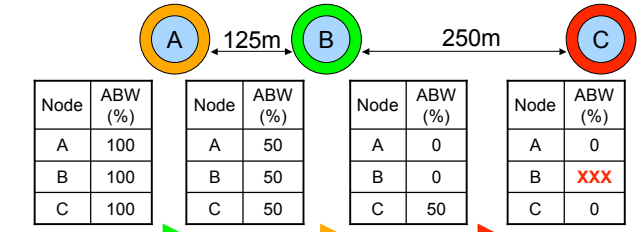
- Many existing QoS solutions are problematic in wireless mobile networks
- Wireless characteristics
  - Shared medium
  - Spatial location
  - Coordination without communication
- Admission control
  - Determining the available bandwidth
- Previous approach drawbacks

Outline

- Motivation
- Background
  - Available bandwidth calculation
  - Wireless communication
- Related work
- PAC: Perceptive Admission Control
- Experimental results
- Conclusion

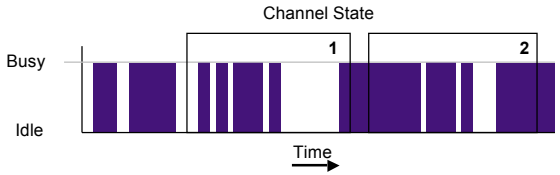
Available Bandwidth (ABW)

- For a new flow to be admitted, nodes need to determine if enough capacity is available
- Reception knowledge is not sufficient



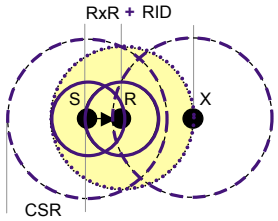
Available Bandwidth Channel Status

- Monitor the channel state - busy/idle
  - Reception range is not enough
  - Use carrier-sensing mechanism
- The more the channel is busy, the higher the utilization
  - Less bandwidth available



Wireless Communication

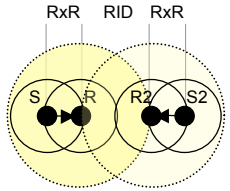
- Reception range
- Carrier-sensing range
- Receiver interference distance
  - Distance between receiver and another source
- Simultaneous transmissions
  - Distance between sources
  - Beyond RxR+RID no impact
- Assumptions
  - Wireless disk model
  - No obstacles



RxR: Reception Range
CSR: Carrier-Sensing Range
RID: Receiver Interference Distance

Wireless Communication

- MAC layer acknowledgements
  - Occur immediately following data reception
  - Receivers do not check CS prior to sending ACKs
- A receiver may now interfere with another receiver
- To prevent interference, two receivers must be at least RID from each other
- Distance between two sources may be as large as 2*RxR+RID



Related Work

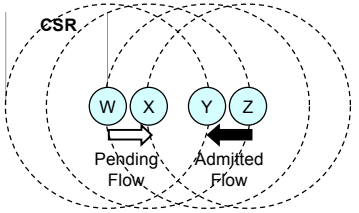
- Stateful
  - RSVP [Zhang '93]
  - INSIGNIA [Lee '00]
- Stateless
  - SWAN [Ahn '02]
  - CACP: Contention-aware Admission Control Protocol [Yang '03]

CACP - Admission Decision

- Check local available bandwidth
- Broadcast admission request message to all neighbors in carrier-sensing range
  - Using high power
- Each Carrier-Sensing Neighbor performs local admission control
  - If a node cannot accommodate the flow, it sends a rejection message back to the requesting node and the session is not admitted
  - Otherwise, after a timeout, the flow is admitted

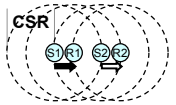
CACP Example

- T_0: Initial state
  - 50% utilized by Z
- 25% pending flow at W
  - T_0: Check
  - T_0: Query
  - Timeout
  - Admit
  - T_1: Adjust
- 50% pending flow at W
  - T_1: Check
  - Query
  - X & Y send rejection messages
  - Reject

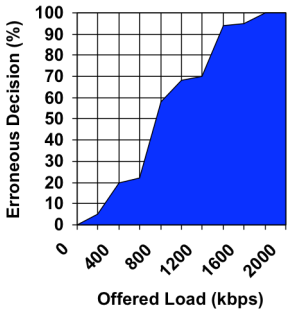


Node	W	X	Y	Z
ABW at T_0	100%	50%	50%	50%
ABW at T_1	75%	25%	25%	50%

Limitations of CACP



- Admission based on rejection messages
  - **When rejection messages are lost, flows are admitted**
  - As the load in the network increases, rejection messages are more likely to be lost, so more flows are likely to be admitted!
  - Since collision rate at R1 is directly proportional to load of S1-R1 flow, error increases with load

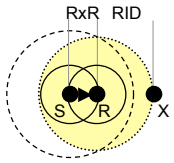


Perceptive Admission Control (PAC)

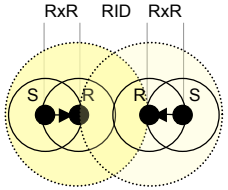
- Nodes monitor channel status within the range that impacts transmission and reception to calculate the available bandwidth
  - RxR+RID or 2*RxR+RID
- If enough bandwidth is available, new traffic can immediately be admitted without querying other nodes
- Available bandwidth measure automatically adjusts to newly admitted traffic
- Congestion can be detected

PAC Measurement Range

- For simultaneous successful transmissions source separation must be at least RxR+RID



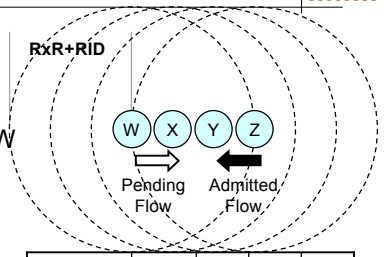
- If ACKs are used, sources must be separated by 2*RxR+RID



RxR: Reception Range
RID: Receiver Interference Distance

PAC Example

- PAC sensing range
- Initial state - T_0
  - 50% utilized by Z
- 25% pending flow at W
  - T_0: Check
  - Admit
  - T_1: Adjust
- 50% pending flow at W
  - T_1: Check
  - Reject



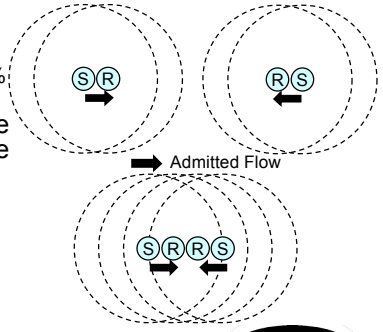
Node	W	X	Y	Z
ABW at T_0	50%	50%	50%	50%
ABW at T_1	25%	25%	25%	25%

Detecting Congestion

- Each node continually monitors available bandwidth
- If available bandwidth drops below a threshold, traffic flows must be throttled or rejected
  - Depends on the type of application
- Conservative bandwidth estimate or bandwidth reservation can help avoid congestion due to mobility, but this decreases the bandwidth utilization

Mobility Example

- Two flows are admitted far from each other
  - Each consumes 75% of network capacity
- When sources move within sensing range of each other, they detect pending congestion and backoff



Evaluation Goals

- Ensure PAC keeps the network from becoming congested
  - Even congestion that is due to mobility
- Multimedia support
  - Packet loss should be small
  - Delay should be low
  - All admitted flows should receive comparable service

Simulation Parameters

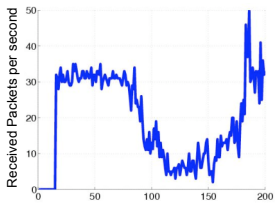
- 50 nodes, 25 sender-receiver pairs
  - Single-hop communication
  - 0-5 m/s and 20 second pause time
- Every 5 seconds another pair starts a CBR traffic flow
  - With PAC, a source only send packets if its flow is admitted
- After 125 seconds of simulation, all senders are active

Overall Performance

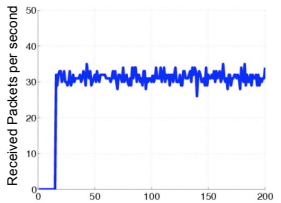
Admission Control	Packet Losses	Packets Delivered	Average Delay (sec)
None	26778	81825	.973
PAC	0	58173	.005

- PAC has no packet losses
- PAC has fewer packets delivered
  - Reserved bandwidth
- PAC has much lower delay

Performance - Goodput



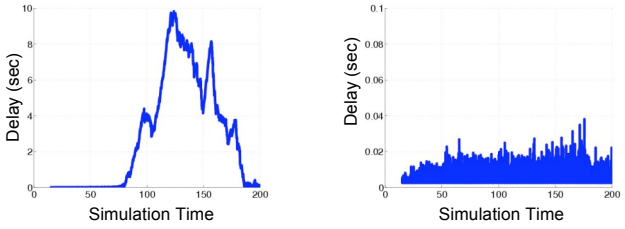
Without Admission Control



With PAC

Throughput of a single receiver

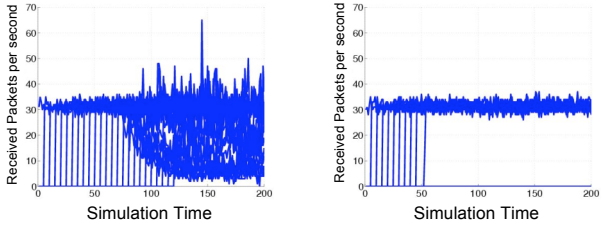
Performance - Delay



Without Admission Control | With PAC

Delay of a single receiver

Performance - Fairness



Without Admission Control | With PAC

Throughput for all flows

Conclusions

- PAC maintains high packet delivery and low delay for all admitted flows
- Simple implementation
- Passive soft-state available bandwidth calculation
- Congestion detection
  - Mobility
- Future work
  - Mote implementation
  - Multiple traffic classes
  - Multihop

Thank You

Questions - Comments?

Ian Chakeres
idc@engineering.ucsb.edu
http://moment.cs.ucsb.edu