Wireless MAC Protocols and TCP Performance

CMU CS 15-744: Computer Networks

Brad Karp

bkarp@cs.cmu.edu

17th October, 2002

Why are we here?

Learn fundamental problems in wireless networking

- How to share wireless medium efficiently and fairly?
- How does TCP behave over wireless links? What can we do to improve its performance?

Learn about designs of systems that are widely used today

Learn to think *critically* about quality of research papers, so you can do good research yourself; acquire taste

Ground rules:

- Feel free to criticize or defend a paper, or my take on it!
- Any comment can lead to discussion!
  (But I reserve right to keep us moving; lots to cover.)

Things to Ask When Evaluating a Paper

Does the paper consider an important, relevant problem?

Does it make reasonable assumptions and use reasonable models?

The longer ago the paper published, the more you should judge if the paper made an impact on the field:

- Does everyone now use systems derived from it, or did they?
- If the paper argued for the importance of trends, did they occur, and did they matter?

Recent papers: judge more on cleverness of ideas, future promise

Old papers: judge on lasting contribution

Other contributions possible: thorough investigation of complex phenomena; comprehensive comparison that brings sense to an area

Wireless Systems: Classes of Network

Cellular Systems [not a topic we'll cover directly]:

- One wireless hop, centralized (mobile to base station)
- Session mobility: call survives changing of base station
- User mobility: user reachable by fixed address (phone number)
- Voice, data
- Many requirements similar to those of mobile IP [next Tuesday]

Wireless LANs [most of what we'll talk about today]

- Base stations
- Peer-to-peer, sometimes multi-hop

Satellite data networks [not a topic we'll cover directly]

Fundamentals: Spectrum and Capacity

A particular radio transmits over some range of frequencies; its bandwidth, in the physical sense

When we've many senders near one another, how do we allocate spectrum among senders? Goals:

- Support for arbitrary communication patterns
- Simplicity of hardware
- Robustness to interference

Shannon's Theorem: there's a fundamental limit to channel capacity over a given spectrum range: $C = B \log_2 (1 + S/N)$

C = capacity (bits/s), B = bandwidth (Hz), S/N = signal/noise power ratio (dB)

Multiple simultaneous senders OK, but no free lunch!

Single-Channel *vs.* Multi-Channel

Suppose we've 100 MHz of spectrum to use for a wireless LAN

Multi-channel wireless:

- Subdivide into 50 channels of 2 MHz each: FDMA, narrow-band transmission
- Radio hardware simple, channels don't mutually interfere
- Multi-path fading (mutual cancellation of out-of-phase reflections)
- Base station can allocate channels to users. How do you support arbitrary communication patterns?
- Other possibilities: FHSS

Single-Channel *vs.* Multi-Channel, cont'd

Single-channel simplex wireless:

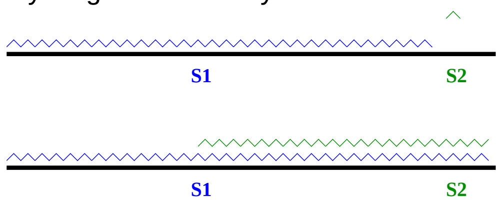
- Spread transmission across whole 100 MHz of spectrum
- Robust to multi-path fading (some frequencies arrive intact)
- Simple: symmetric radio behavior
- Supports peer-to-peer communication
- Collisions: a receiver must only hear one strong transmission at a time

Review: Ethernet's CSMA/CD

"Ethernet is straight from God." - H.T. Kung, in networks course

CS (Carrier Sense): listen for others' transmissions before transmitting; defer to others you hear

CD (Collision Detection): while you transmit, listen and verify you hear exactly what you send; if not, back of a random interval, over an exponentially longer each time you transmit unsuccessfully



What does CSMA/CD require to work correctly (catch all collisions) on a link? Is CD possible on a wireless link? Why or why not?

MACAW Context

Published in SIGCOMM 1994, work 93-94

802.11 standardization proceeded in parallel (IEEE standard in 1997)

MACAW and 802.11 similar; both draw heavily from Karn's MACA

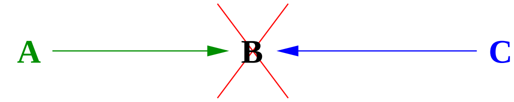
No real research paper on 802.11 design; MACAW covers same area well

Assumptions: uniform, circular radio propagation; fixed transmit power

What are authors' stated goals?

- Fairness in sharing of medium
- Efficiency (total bandwidth achieved)
- Reliability of data transfer at MAC layer

Hidden Terminal Problem



Nodes placed a little less than one radio range apart

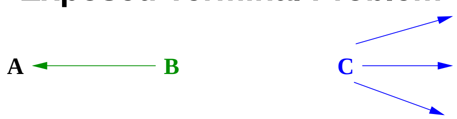
CSMA: nodes listen to determine channel idle before transmitting

C can't hear A, so will transmit while A transmits; result: collision at B

Carrier Sense insufficient to detect all transmissions on wireless networks!

Key insight: collisions are spatially located at receiver

Exposed Terminal Problem



Two flows, this time B sends to A; C sends to a node other than B

If C transmits, does it cause a collision at A?

Yet C will refuse to transmit while B transmits to A!

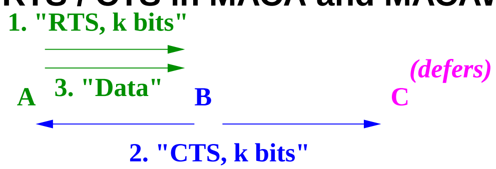
Same insight: collisions are spatially located at receiver

Thinking ahead: implications for multi-hop forwarding?

One possibility: directional antennas (see Mobicom 2002) rather than omnidirectional antennas. Why does this help? Why is it hard?

Simpler solution: use *receiver's* medium state to determine transmitter behavior

RTS / CTS in MACA and MACAW



Sender sends short, fixed-size RTS packet to receiver

Receiver responds with CTS packet

RTS and CTS both contain length of data packet to follow from sender

Solves hidden terminal problem!

Absent CTS, sender backs off exponentially (BEB) before retrying

RTS and CTS can still themselves collide at their receivers; less chance as they're short; any help on short data packets?

What's the effect on exposed terminal problem?

BEB in MACA

Current backoff constant: B

Maximum backoff constant: B_M

Minimum backoff constant: B_0

MACA sender:

- $B_0 = 2$ and $B_M = 64$
- Upon successful RTS/CTS, $B \leftarrow B_0$
- Upon failed RTS/CTS, $B \leftarrow \min[2B, B_M]$

Before retransmission, wait a uniform random number of RTS lengths (30 bytes) in $[0, B]$

No carrier sense! (Karn concluded useless because of hidden terminals)

BEB in MACAW

BEB can lead to unfairness: backed-off sender has decreasing chance to acquire medium ("the poor get poorer")

Simple example: two senders sending to the same receiver, each sending at a rate that can alone saturate the network

MACAW proposal: senders write their B into packets; upon hearing a packet, adopt its B

Result: dissemination of congestion level of "winning" transmitter to its competitors

Is this a good idea?

RTS failure rate at one node propagates far and wide

Ambient noise? Regions with different loads?

Reliability: ACK

MACAW introduces an ACK after DATA packets; not in MACA

Sender retransmits if RTS/CTS succeeds but no ACK returns; doesn't back off

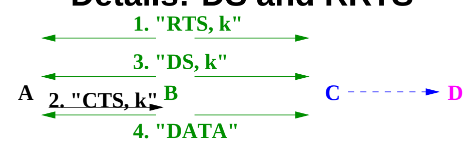
Rapid loss recovery, as compared with TCP (compare RTT on LAN to WAN)

Useful when there's ambient noise (microwave ovens . . .)

Why are sequence numbers in DATA packets now important (not mentioned directly in paper!)

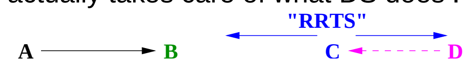
Are ACKs useful for multicast packets? Consequences for, *e.g.*, ARP?

Details: DS and RRTS



In exposed terminal problem, how does C know not to send RTSes, and grow its backoff?

Carrier sense actually takes care of what DS does . . .



Once A wins in contention, its large data packets don't give C a chance to send a CTS to D!

MACAW fix: C "proxy contends" for D, by sending an RRTS packet

Now reverse A-B flow to B-A. RRTS no help. C can't hear D's RTS packets!

MACAW and 802.11 Differences

802.11 uses physical CS before transmissions: defers a uniform random period, in $[0, B]$

802.11 combines physical CS with virtual CS from RTS/CTS packets in the Network Allocation Vector (NAV)

802.11 uses RTS-CTS-DATA-ACK

802.11 uses BEB when an ACK doesn't return

802.11: A Dose of Reality

The canonical wireless link in the research community. Why?

- Hardware commoditized, cheap
- First robust (DSSS) wireless network with LAN-like bitrate

Many, many wireless system papers based on ns simulations of 802.11 networks

Caveat simulator: simulating a real link layer doesn't mean realistic simulations. Interference models? Traffic patterns? Mobility patterns?

Have I been wasting your time? In practice *no one uses RTS/CTS!!*

Why? Are MACAW and the hidden terminal problem irrelevant?

Traffic Workloads and Hidden Terminals

To first order, everyone uses base stations, not peer-to-peer 802.11 networks

When base station transmits, there can be no hidden terminals. Why?

Clients can be hidden from one another. But what's the average packet output stream of a wireless client? Packet sizes?

What's the cost of RTS/CTS? How big are RTS and CTS packets?

802.11 end-user documentation recommends *disabling* RTS/CTS "unless you are experiencing unusually poor performance"

Drivers leave it off by default

RTS/CTS still very valuable for peer-to-peer workloads

TCP over Wireless: Background

On the wired Internet, packets are dropped by congested routers

TCP's AIMD congestion control responds to congestion by halving window; stable, avoids driving network into congestive collapse

What if packets are dropped for other reasons?

- Environmental noise causes packet corruption
- Weak signal (nearly out-of-range) causes corrupted packets

Fundamental: How does TCP determine loss? Does feedback distinguish between congestion and corruption?

Architectural questions: Where do we fix it? Link layer? In TCP itself? Both? Do link layers need to look inside TCP packets?

What are advantages and disadvantages of these architectural approaches?

TCP over Wireless [BPS+97] Context

Written in '97, work spans '94 - '97

For TCP-related work, what's deployed matters; at time of paper's writing Reno was everywhere, NewReno novelty; now NewReno everywhere

SACK is also widely deployed now (one vendor matters a lot :-))

See TBIT [Padhye, Floyd] in SIGCOMM 2001 on changing state of TCP deployment

What are authors' stated goals?

- Explore complicated space of alternatives for TCP over wireless; measure relative performance (quantitative).
- Show detailed interactions between TCP and these alternatives (qualitative).

Classes of TCP Enhancements

End-to-end:

- Reno, NewReno: NewReno responds to partial ACKs by sending one new packet per RTT, when there are multiple losses in a window
- SACK: in addition to cumulative ACK, receiver returns SACK blocks which identify non-contiguously received data above the cume ACK
- Explicit Loss Notification (ELN): when wireless receiver detects a frame is corrupted, marks future ACKs for that connection that loss was corruption, not congestive loss

Classes (cont'd)

Link-Layer:

- LL: cume ACKs at link layer, retransmission after 200 ms; uses TCP ACKs to determine when to retransmit

Compare with MACAW, 802.11

What effect does LL introduce of note to the TCP sender?

LL-TCP-AWARE addreses this effect by hiding dupacks from TCP sender; identical to Snoop protocol—cache packets at wired/wireless gateway, locally retransmit

Split-connection protocols:

- SPLIT: Terminate TCP connection at wired/wireless gateway host

Performance Results Capsule

LL-TCP-AWARE offers best throughput across varying bit error rates

Next lowest throughput is E2E-SMART, which outperforms E2E-IETF-SACK, but assumes no reordering

Is TCP-AWARE necessary??

More general technique: make sender robust to dupacks!

Useful in making TCP robust to reordering [RRTCP, 2002]

Link model validation?