MACA: A New Channel Access Method for Packet Radio

the proceedings of the 9th ARRL Computer Networking Conference, London, Ontario, Canada, 1990.

Author: Phil Karn

Goals , New Ideas, and Main Contributions

- Goals:
  - Try to overcome hidden & exposed terminal problems
- New idea:
  - Reserve the channel before sending data packet
  - Minimize the cost of collision (control packet is much smaller than data packet)
- Main Contribution:
  - A three-way handshake MAC protocol : MACA

CSMA/CA MA/CA MACA

Fundamental Assumptions

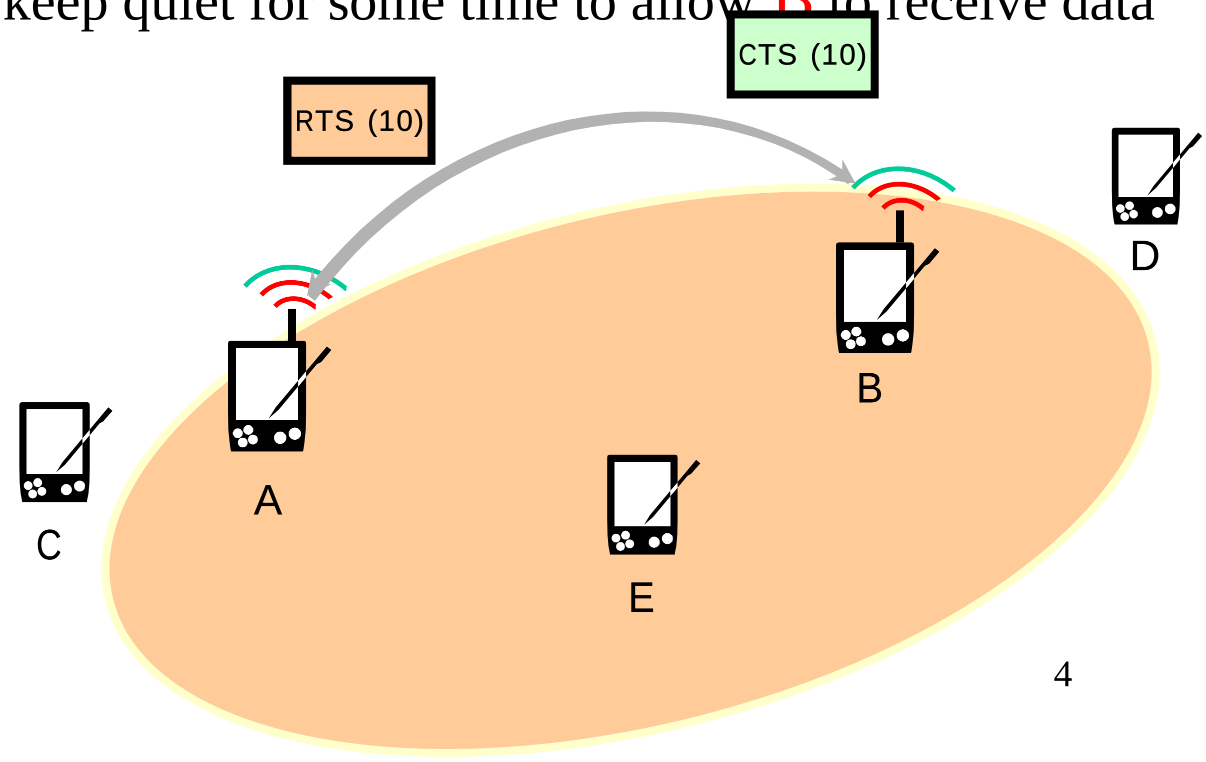
- Symmetry
  - A can hear from B ó B can hear from A
- No capture
- No channel fading
- Packet error only due to collision
- Data packets and control packets are transmitted in the same channel

Three-Way Handshake

- A sends Ready-to-Send (RTS)
- B responds with Clear-to-Send (CTS)
- A sends DATA PACKET
- RTS and CTS announce the duration of the data transfer
- Nodes overhearing RTS keep quiet for some time to allow A to receive CTS
- Nodes overhearing CTS keep quiet for some time to allow B to receive data packet

CTS: Clear To Send

RTS: Request To Send



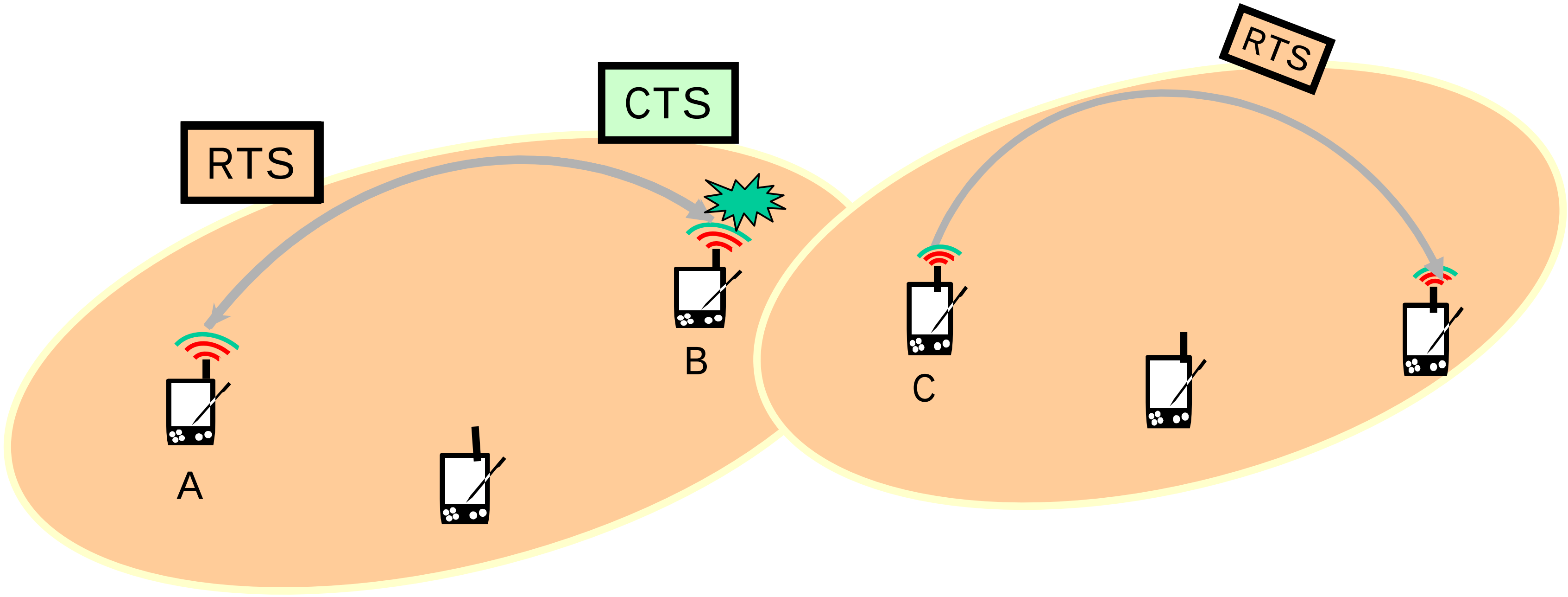
More Details for MACA

- A sends out RTS and set a timer and waits for CTS
  - If A receives CTS before timer go to zero, OK! sends data packet
  - Otherwise, A assumes there is a collision at B
    - Double the backoff counter interval
      - Randomly pick up a timer from [1,backoff counter]
    - Send next RTS after timer go to zero
- B sends out CTS, then set a timer and waits for data packet
  - If data packet arrives before timer go to zero, OK!
  - Otherwise, B can do other things
- C overhears A's RTS, set a timer which is long enough to allow A to receive CTS. After the timer goes to zero, C can do other things
- D overhears B's CTS, set a timer which is long enough to allow B to receive data packet.
- E overhears A's RTS and B's CTS, set a timer which is long enough to allow B to receive data packet.
- RTS and CTS can also contain info to allow sender A to adjust power to reduce interference

Note: no carrier sense

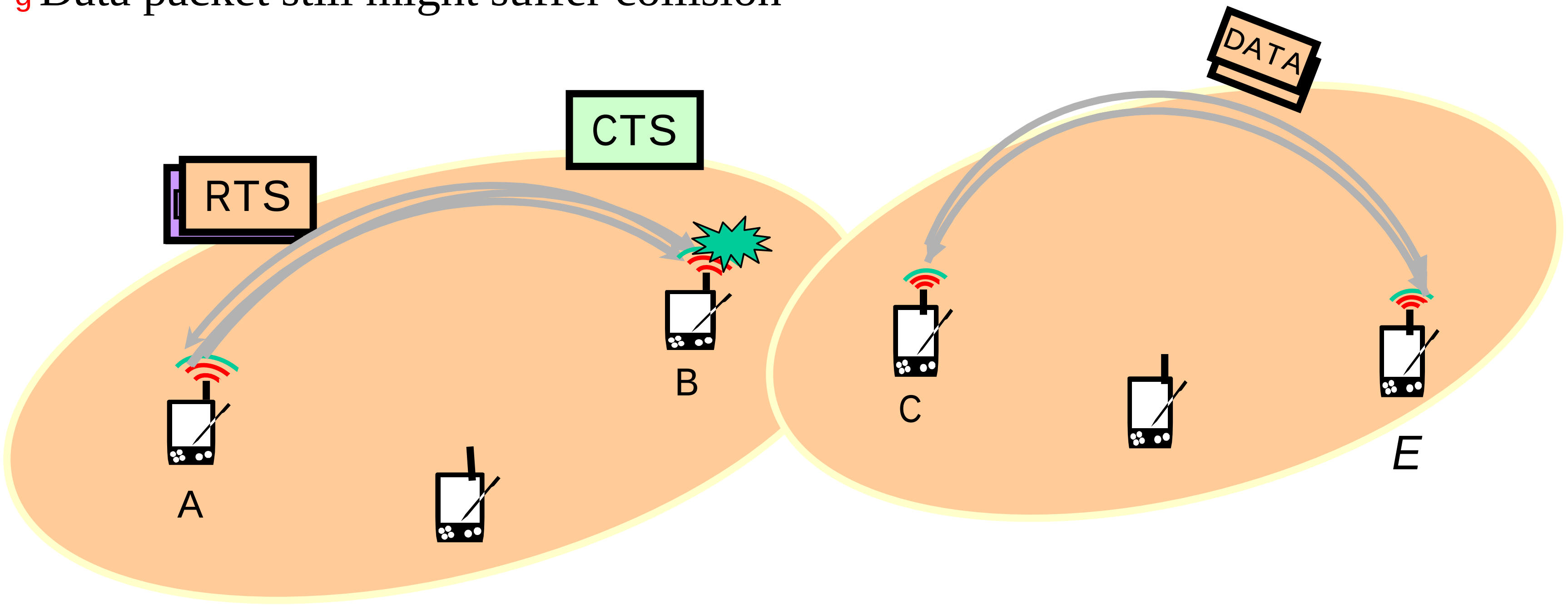
Hidden Terminal Problem Still Exists (1)

- Data packet still might suffer collision



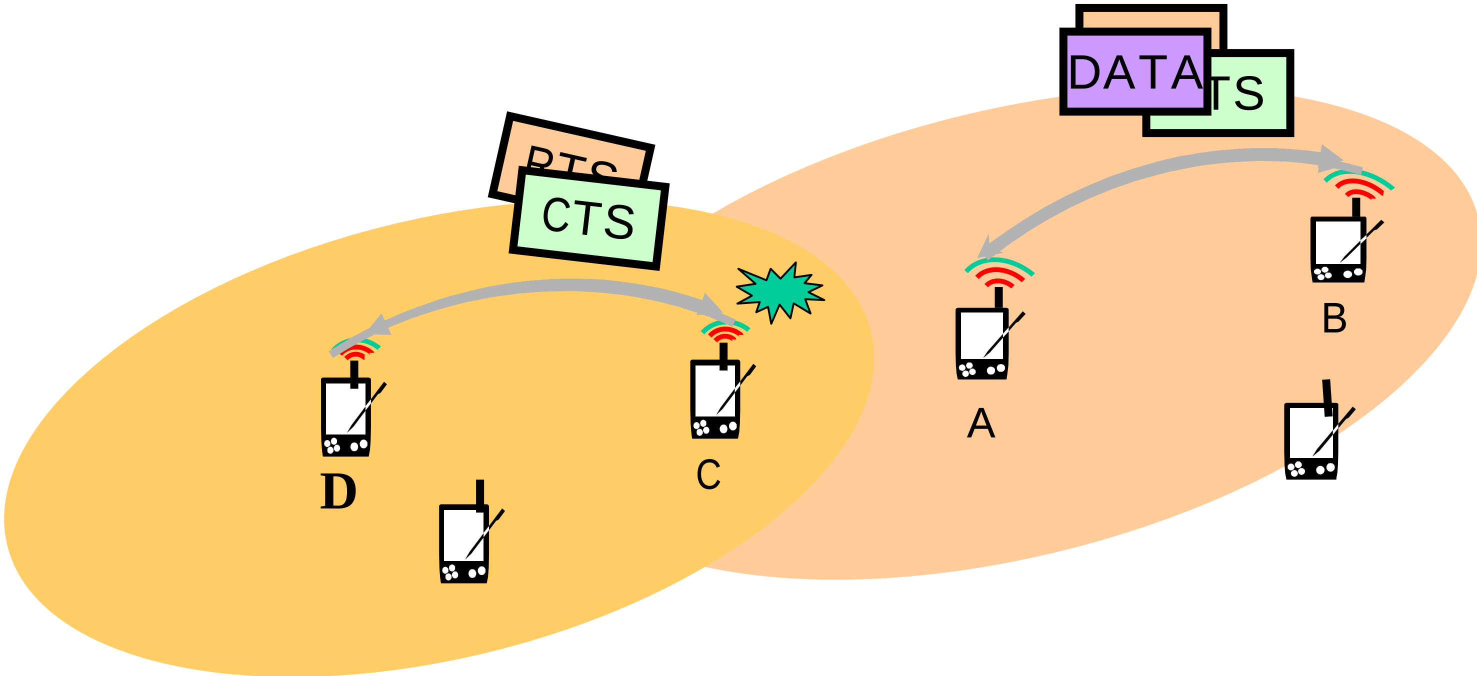
Hidden Terminal Problem Still Exists (2)

- Data packet still might suffer collision



Exposed Terminal Problem Still Exists

- Node C can not receive CTS



Summary

- MACA did not solve hidden & exposed terminal problems
- MACA did not provide specifications about parameters
  - What are RTS, CTS packet sizes ?
  - How to decide timers?
  - What is initial backoff window size?
- A lot things need to do if using MACA

MACAW: A Media Access Protocol for Wireless Lan’s

ACM Sigcomm '94, London, UK.

V. Bharghavan, A. Demers, S. Shenker, and L. Zhang (Sigcomm 1994)

Goals, New Ideas, and Main Contributions

- Goals:
  - This paper refined and extended MACA
  - Improve fairness and increase throughput
- New Idea: Information sharing to achieve fairness
- Main Results:
  - Modified control messages
    - Four-way handshake (reliable, recover at MAC layer)
    - Five-way handshake (relieve exposed terminal problem)
    - RRTS (unfairness)
  - Modified back-off algorithms
    - Multiplicative increase and linear decrease (MILD)
    - Synchronize back-off counter using piggyback message
  - Multiple stream model (V-MAC)

Revisit Hidden Terminal Problem

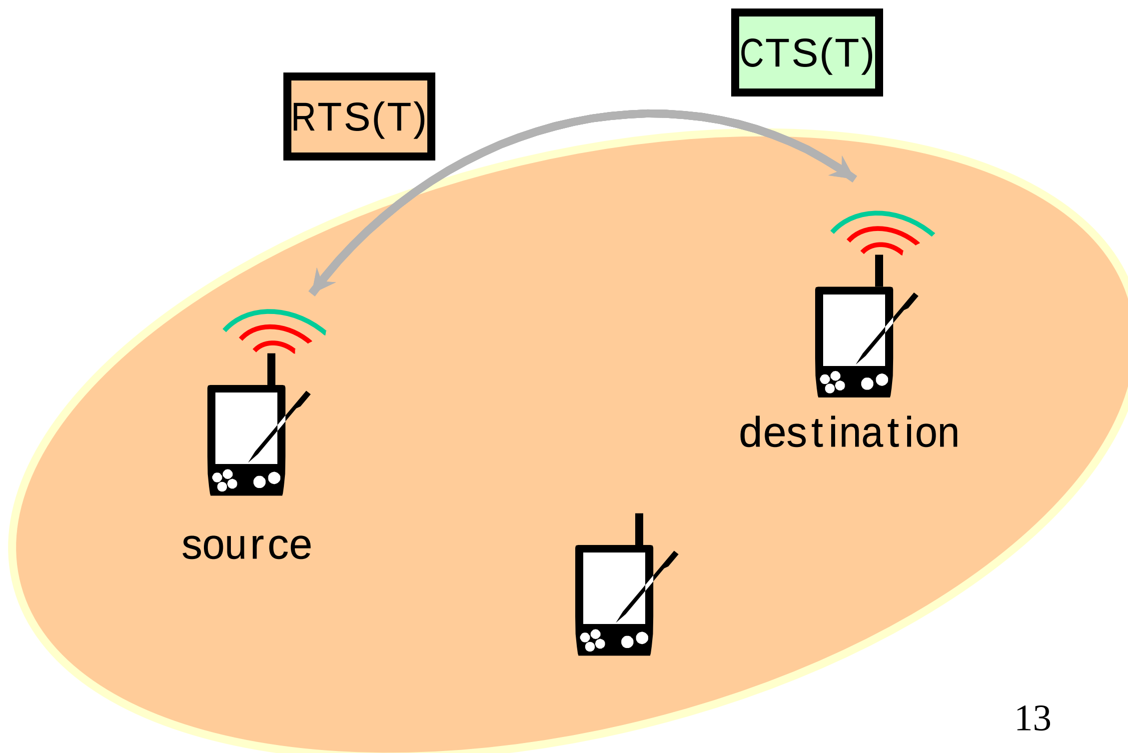
- Data packet still may suffer collision
- To recover packet loss at transport layer is too slow
- Recover at MAC layer is faster
- Need ACK from destination

Four-Way Handshake

- Sender sends Ready-to-Send (RTS)
- Receiver responds with Clear-to-Send (CTS)
- Sender sends DATA PACKET
- Receiver acknowledge with ACK
- RTS and CTS announce the duration of the transfer
- Nodes overhearing RTS/CTS keep quiet for that duration
- Sender will retransmit RTS if no ACK is received
  If ACK is sent out, but not received by sender, after receiving new RTS, receiver returns ACK instead of CTS for new RTS

CTS: Clear To Send

RTS: Request To Send



Comparison with ACK and without ACK

Error Rate	RTS-CTS-DATA	RTS-CTS-DATA-ACK
0	40.41	36.76
0.001	36.58	36.67
0.01	16.65	35.52
0.1	2.48	9.93

Table 4: The throughput, in packets per second, achieved by a single TCP data stream between a pad and a base station in the presence of noise.

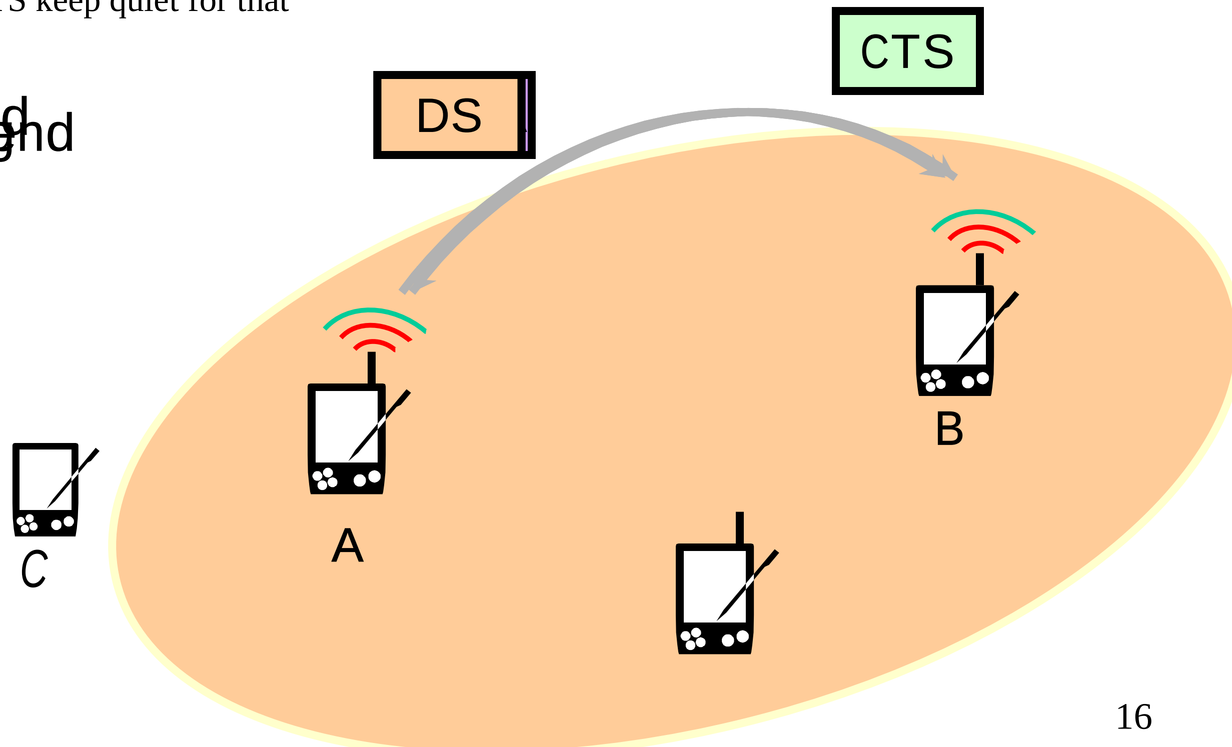
Revisit Exposed Terminal Problem

- RTS/CTS/DATA/ACK can not solve exposed terminal problem
- When overhearing RTS, the node needs to wait longer enough to allow the data packet being completely transmitted even it does not overhear CTS
- To relieve exposed terminal problem,
  - Let exposed terminal know the DATA packet does be transmitted
  - Extra message DS (data send)
- Five Handshaking to let exposed terminal know how long it should wait

Five-Way Handshake

- Sender sends Ready-to-Send (RTS)
- Receiver responds with Clear-to-Send (CTS)
- Sender sends DATA SENDING (DS)
- Sender sends DATA PACKET
- Receiver acknowledge with ACK
- RTS and CTS announce the duration of the transfer
- Nodes overhearing RTS/CTS keep quiet for that duration

RTS: Request To Send
CTS: Clear To Send
DS: Data Sending



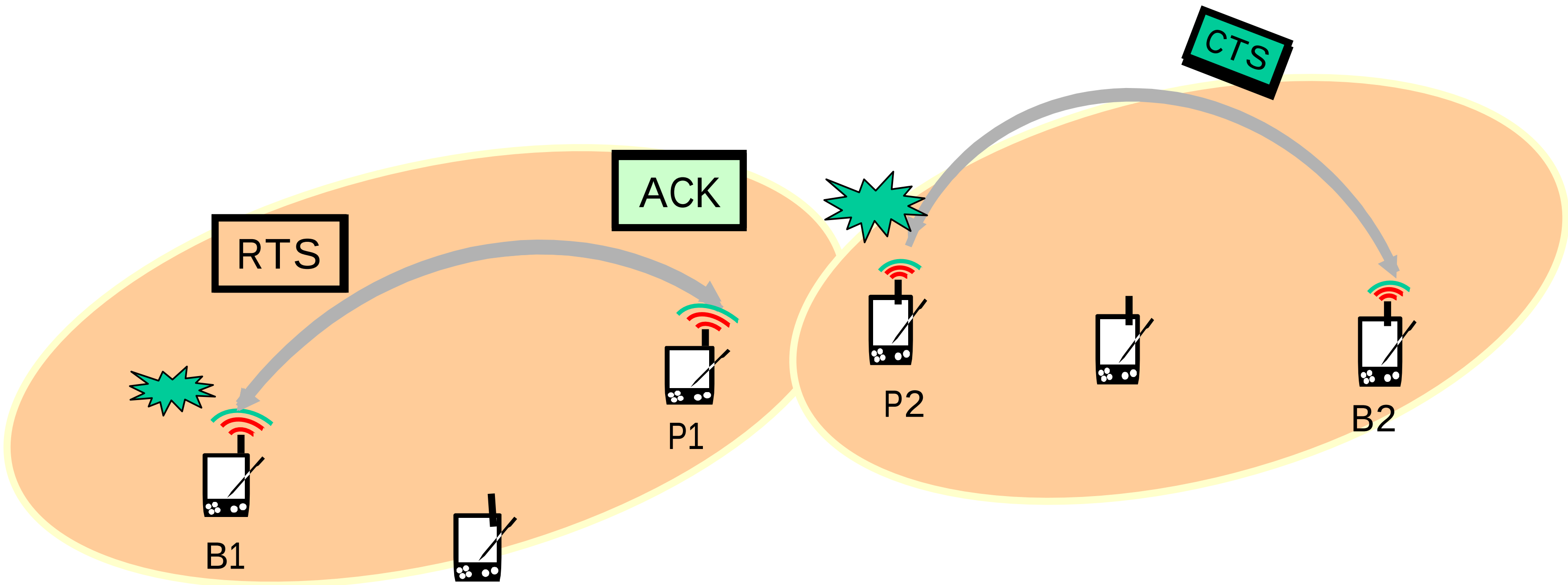
Comparison with DS and without DS

Figure 5: A two cell configuration where both pads are in range of their respective base stations and also in range of each other. The pads are sending data to their base stations, and each stream is generating data at a rate of 64 packets per second and using UDP for transport.

	RTS-CTS-DATA-ACK	RTS-CTS-DS-DATA-ACK
P1-B1	46.72	23.35
P2-B2	0	22.63

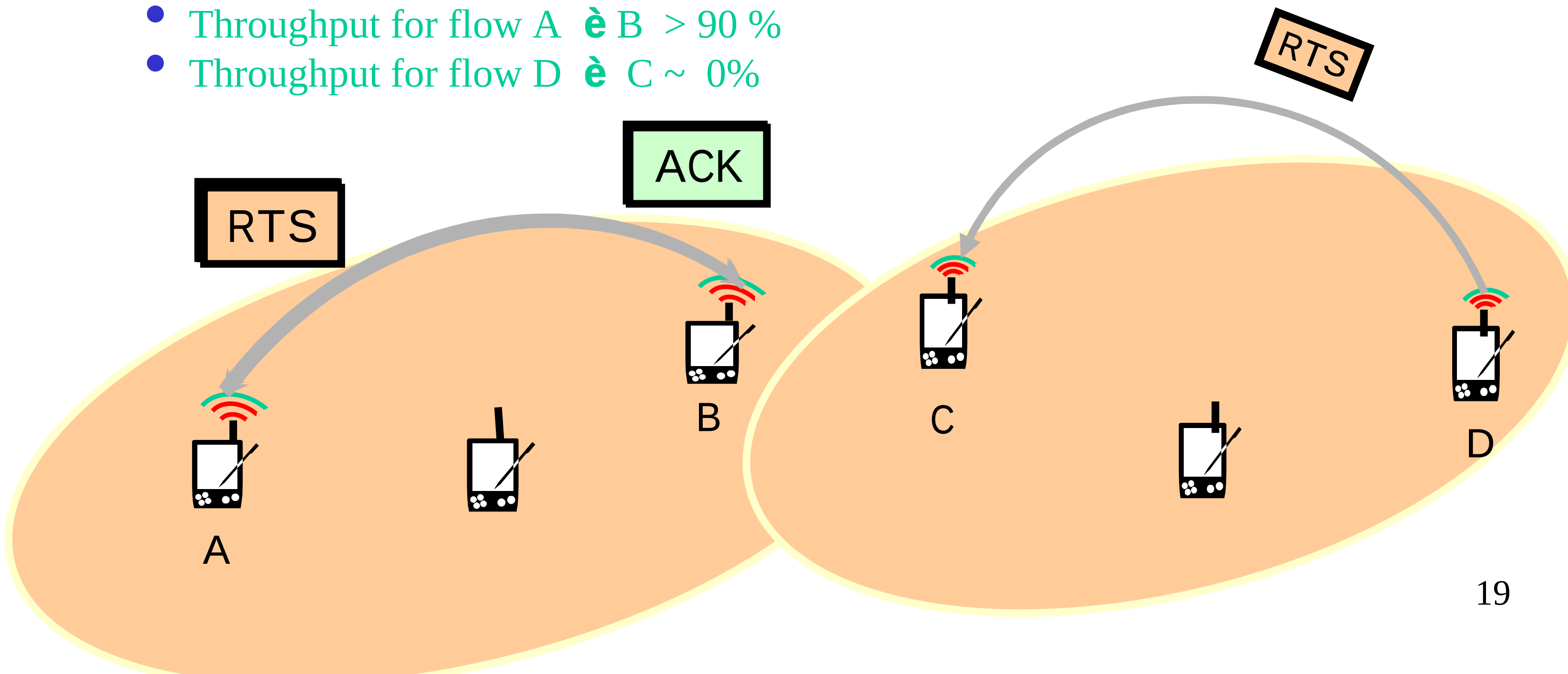
Table 5: The throughput, in packets per second, achieved by the streams in Figure 5.

Comparison with DS and without DS



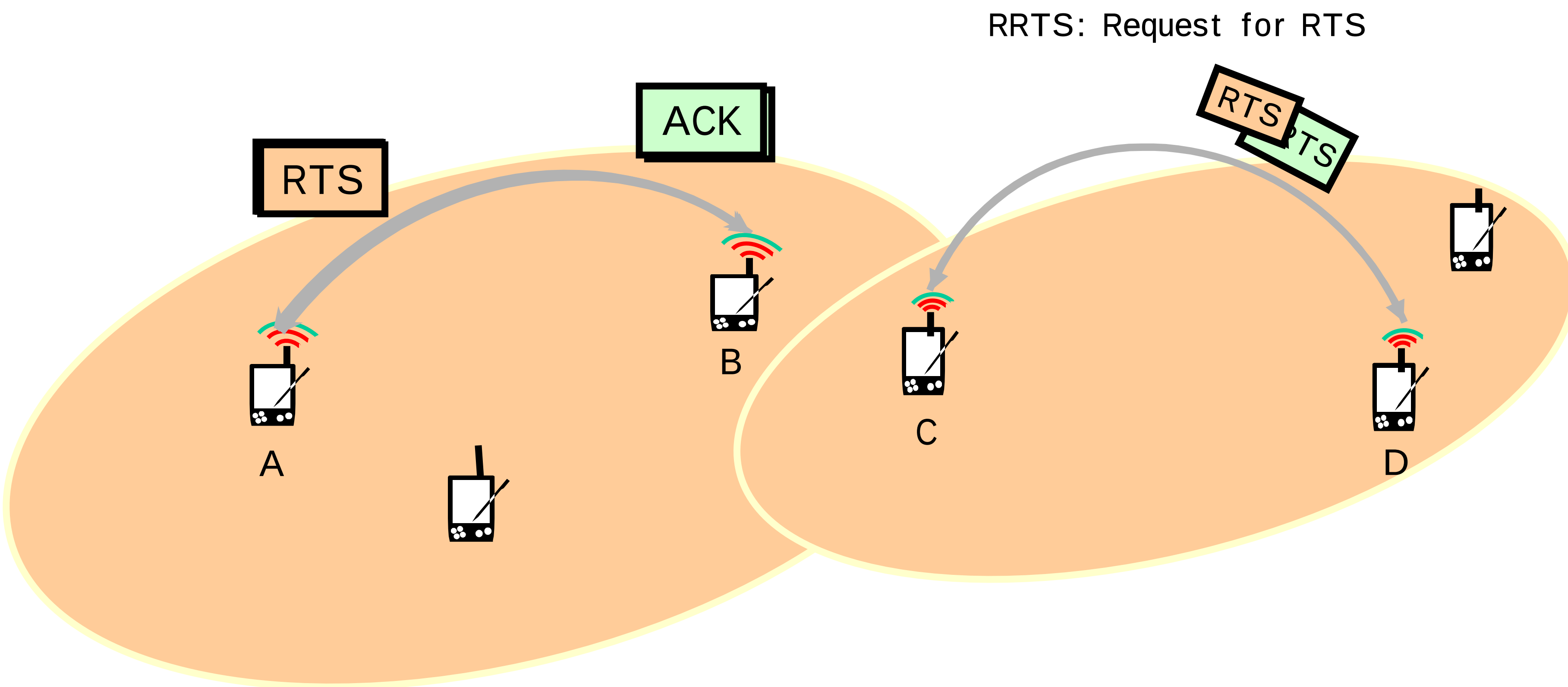
Unfairness

- Using *RTS/CTS/DATA/ACK* or *RTS/CTS/DS/DATA/ACK* might cause unfairness
- A sends data to B; D sends data to C
- A and D have enough data to send
- C can hears from B and D, but not A
- B can hear from A and C, but not D
  - A is in luck and gets the channel
  - D sends RTS and times out
  - Backoff window for D repeatedly doubles
  - For the next transmission:
    - A picks a random number from a smaller window
    - Unequal probability of channel access
    - Throughput for flow A è B > 90 %
    - Throughput for flow D è C ~ 0%



Request for RTS (RRTS)

- Try to solve unfairness by having C do the contending for D



Why Uses RRTS Instead Of CTS ?

- CTS or RTS packet size << data packet size
- When nodes overhear CTS, they need to defer a time period to allow the expected data packet transmission
- When nodes overhear RRTS, they only need to defer a time period to overhear the expected CTS
- Uses CTS will cost long waiting

Comparison with RRTS and without RRTS (1)

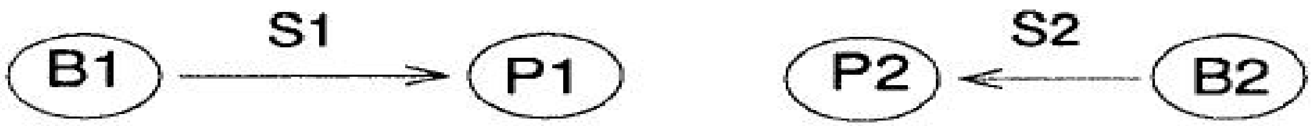


Figure 6: A two cell configuration where both pads are in range of their respective base stations and also in range of each other. The base stations are sending data to their respective pads, and each stream is generating data at a rate of 64 packets per second and using UDP for transport.

	no RRTS	RRTS
B1-P1	0	20.39
P2-B2	42.87	20.53

Table 6: The throughput, in packets per second, achieved by the streams in Figure 6.

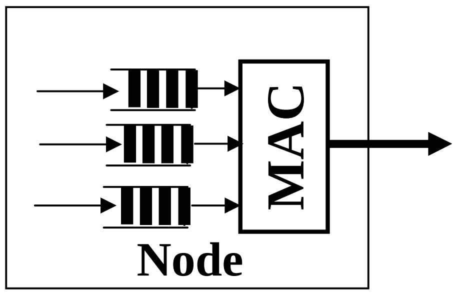
Comparison with RRTS and without RRTS (2)

Figure 7: A two cell configuration where both pads are in range of their respective base stations and also in range of each other. Base station B1 is sending data to pad P1, and pad P2 is sending data to base station B2. Each stream is generating data at a rate of 64 packets per second and using UDP for transport.

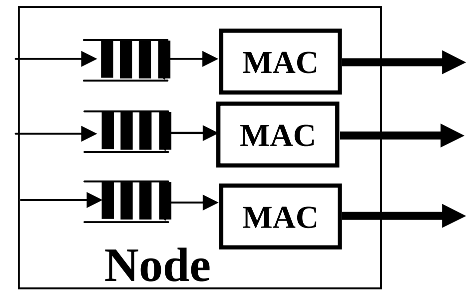
B1-P1	0
P2-B2	43.93

Table 7: The throughput, in packets per second, achieved by the streams in Figure 7.

Multiple Stream Model (V-MAC)



Single Stream MAC



Multiple Stream MAC

- Single stream model merges traffic from different flows into a mixed stream and uses a single MAC
- Multiple stream model uses multiple MAC (one flow one MAC) to achieve fairness
- This idea was used by Intersil Company to propose a new MAC for IEEE 802.11e in 2001

Why Multiple Stream MAC more fair Than Single Stream MAC

- When collision
  - all packets in single stream MAC are used a large backoff window
  - Different flow's packet in multiple stream MAC uses different backoff window

Comparison V-MAC and MAC

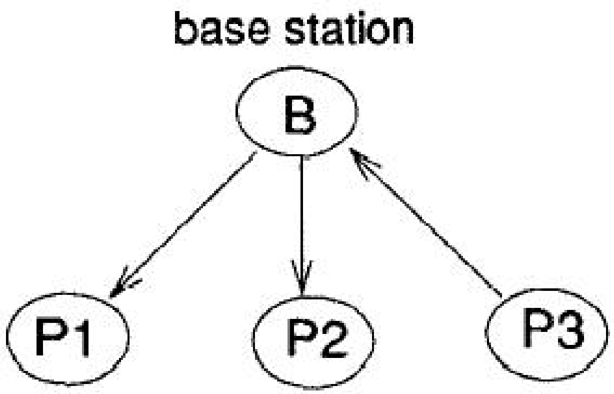


Figure 4: A single cell configuration where all stations are in range of each other. The base station is sending data to two of the pads, and the third pad is sending data to the base station. Each stream is generating data at a rate of 32 packets per second and using UDP for transport.

	Single Stream	Multiple Stream
B-P1	11.42	15.07
B-P2	12.34	15.82
P3-B	22.74	15.64

Table 3: The throughput, in packets per second, achieved by the streams in Figure 4.

Backoff Algorithms

- When collision occurs, node A pick up a random number T from [1,Bo], then retransmits RTS after T time unit

- How to determine Bo
  - After each collision Bo_new = Fun_inc(Bo_old)
  - After each successful transmission Bo_new = Fun_dec(Bo_old)

- Binary exponential backoff (BEB) algorithm
  - Fun_inc(Bo_old)=min{2*Bo_old, Bo_max}
  - Fun_dec(B_old)=Bo_min

- Multiplicative increase linear decease (MILD)
  - Fun_inc(Bo_old)=min{1.5*Bo_old, Bo_max}
  - Fun_dec(B_old)=max{Bo_old -1, Bo_min}

Information Sharing in Backoff Algorithms

- When a node sends a packet, it embeds its current backoff counter in the packet header. Other nodes which overhears the packet copy the value as itself backoff counter
- Key idea: all nodes have the same backoff counter to achieve fairness

Comparison BEB and BEB-Copy

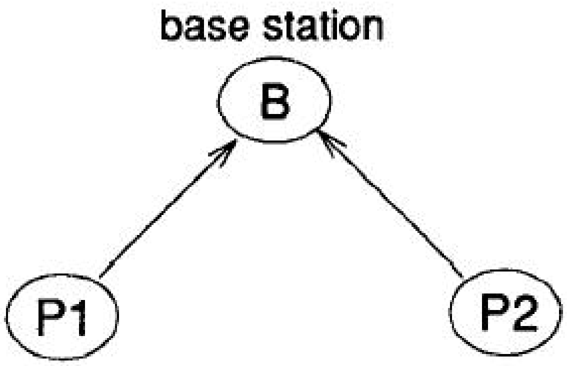


Figure 2: A single cell configuration where all stations are in range of each other and both pads are sending data to the base station (the arrows indicate the direction of the data transmission). The pads are each generating data at a rate of 64 packets per second and are using UDP for transport.

	BEB	BEB copy
P1-B	48.5	23.82
P2-B	0	23.32

Table 1: The throughput, in packets per second, achieved by the streams in Figure 2.

Comparison BEB-COPY and MILD-Copy

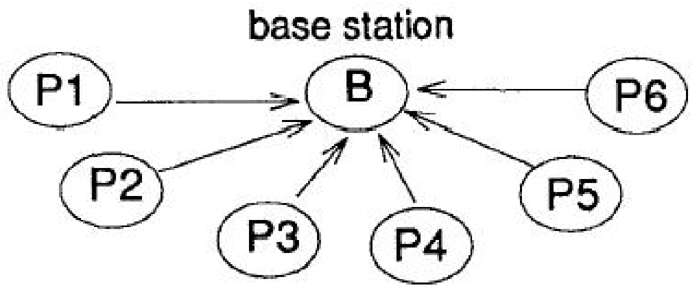


Figure 3: A single cell configuration where all stations are in range of each other. All six pads are sending data to the base station. Each stream is generating data at a rate of 32 packets per second and using UDP for transport.

	BEB copy	MILD copy
P1-B	2.96	6.10
P2-B	3.01	6.18
P3-B	2.84	6.05
P4-B	2.93	6.12
P5-B	3.00	6.14
P6-B	3.05	6.09

Table 2: The throughput, in packets per second, achieved by the streams in Figure 3.

Per-Destination Backoff

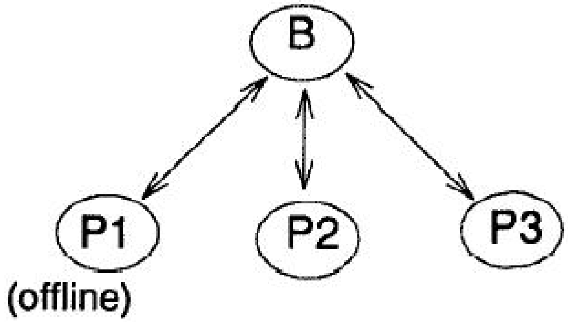


Figure 9: A single cell configuration where all pads are in range of the base stations and also in range of each other. The base station is sending data to each pad, and each pad is sending data to the base station. Each stream is generating data at a rate of 64 packets per second and using UDP for transport. Pad P1 is turned off after 300 seconds.

	Single backoff	Per-destination backoff
B1-P2	3.79	8.98
P2-B1	3.78	8.84
B1-P3	3.62	8.68
P3-B1	3.43	8.41

Table 8: The throughput, in packets per second, achieved by the streams in Figure 9.

Evaluation of MACAW

MACA	RTS-CTS-DATA	53.07
MACAW	RTS-CTS-DS-DATA-ACK	49.07

Table 9: The throughput, in packets per second, achieved by a uncontested single stream.

Evaluation of MACAW

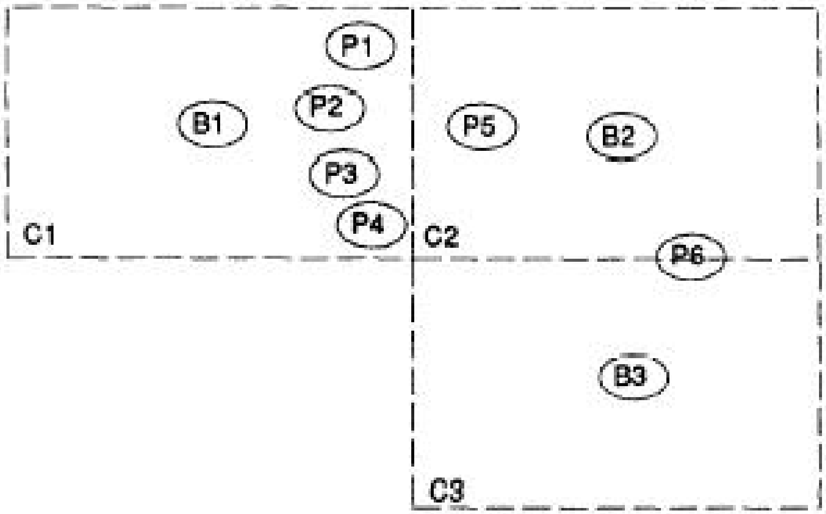


Figure 10: A configuration with three cells with varying levels of congestion.

	MACA	MACAW
P1-B1	9.61	3.45
P2-B1	2.45	3.84
P3-B1	3.70	3.27
P4-B1	0.46	3.80
B1-P1	0.12	3.83
B1-P2	0.01	3.72
B1-P3	0.20	3.72
B1-P4	0.66	3.59
P5-B2	2.24	7.82
B2-P5	3.21	7.80
P6-B3	28.40	25.16

Table 10: The throughput, in packets per second, achieved by the streams in Figure 10.

Every flow has the same data rate
32 packet per second

Total Troughput
MACA: 51.06
MACAW: 70
37% higher

Evaluation of MACAW

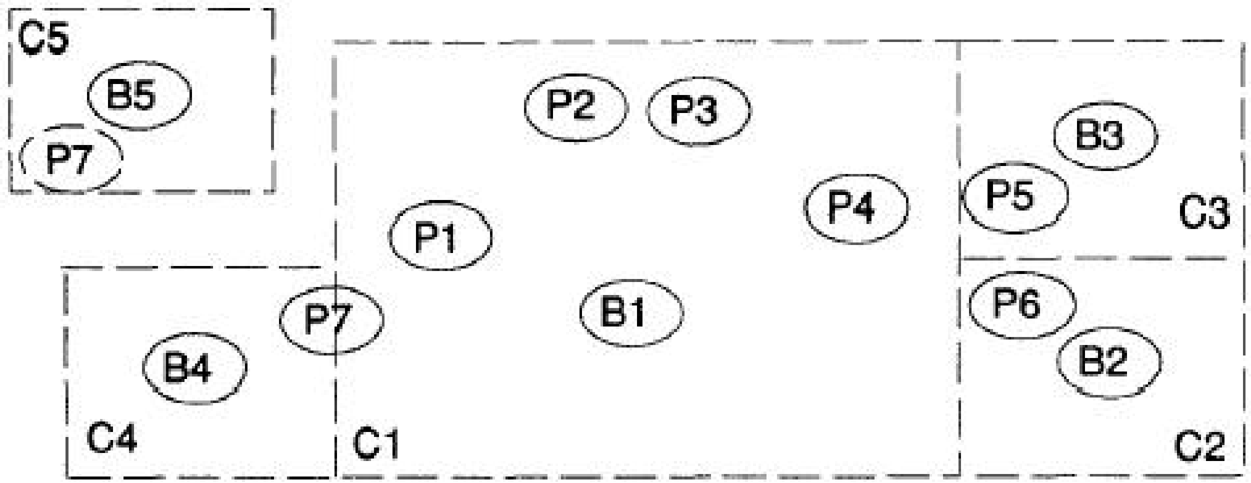


Figure 11: A configuration based on part of the Computer Science Laboratory at PARC.

Evaluation of MACAW

	MACA	MACAW
P1-B1	0.78	2.39
P2-B1	1.10	2.72
P3-B1	0.22	2.54
P4-B1	0.06	2.87
P5-B3	18.17	14.45
P6-B2	6.94	14.00
P7-B4	23.82	19.18

Table 11: The throughput, in packets per second, achieved by the streams in Figure 11.

Open Problems

- How to design a good backoff algorithm?
- Adaptive MAC to achieve fairness in ad-hoc networks
- Do upper layer operations need to tightly relate to MAC?
- Reliable multicast MAC in ad-hoc networks