MAC3: Medium Access Coding & Congestion Control

Devavrat Shah (MIT)

Damon Wischik (UCL)

Information theory and coding theory

"What is the theoretical limit on how much I can send?"

full knowledge — coding — noisy channels

Can we use coding to deal with contention?

While adapting to an ever-changing network?

And achieve an optimal allocation of capacity?

Wireless MAC

"How can I learn when it's safe to send?"

contention — adapting — distributed algorithm

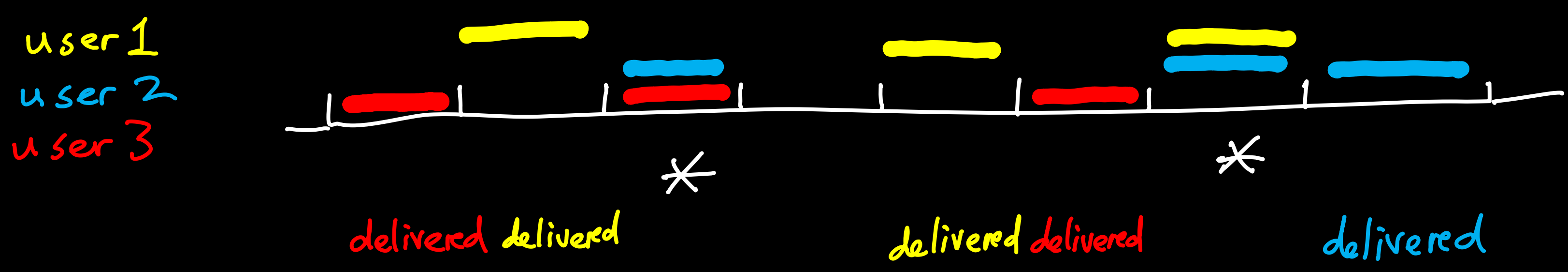
Congestion control

"How can I learn how fast to send?"

adapting — distributed algorithm — resource allocation — networks

The classic ALOHA model for a wireless channel is useful for studying contention & coordination.

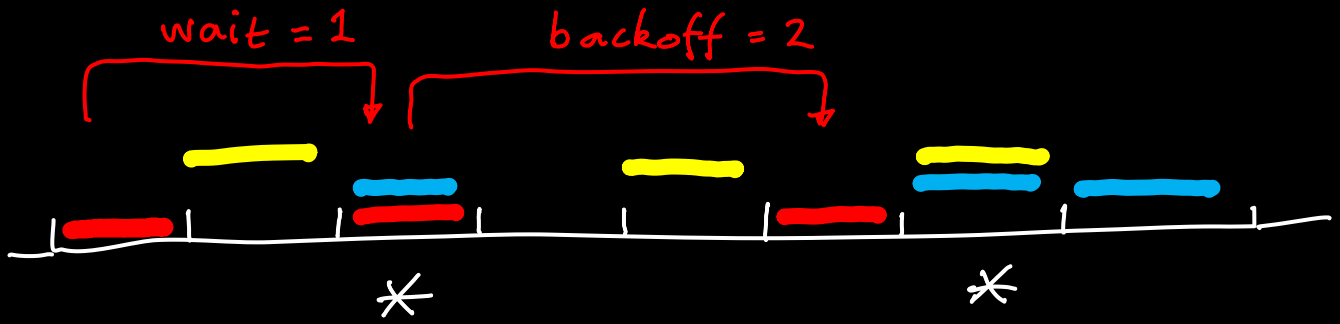
- Let there be *n* users, each with an infinite backlog of packets. Over a long timescale, *n* varies.
- Time is slotted. In each timeslot, users can transmit a packet.
- If exactly one user transmits, its packet is successfully delivered. If more than one user transmits, the packets collide and none is successfully delivered.



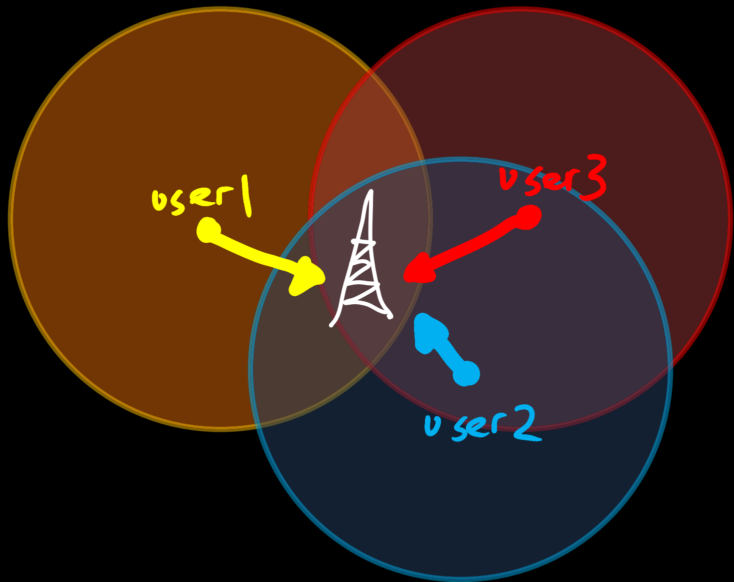
- At the end of the timeslot, all users are told whether there were no attempts, one attempt, or more than one attempt.
- The Medium Access Control (MAC) protocol is the algorithm for deciding whether a user should attempt transmission in a given timeslot. How should the MAC protocol be designed?

What are the typical mechanisms of a MAC protocol?

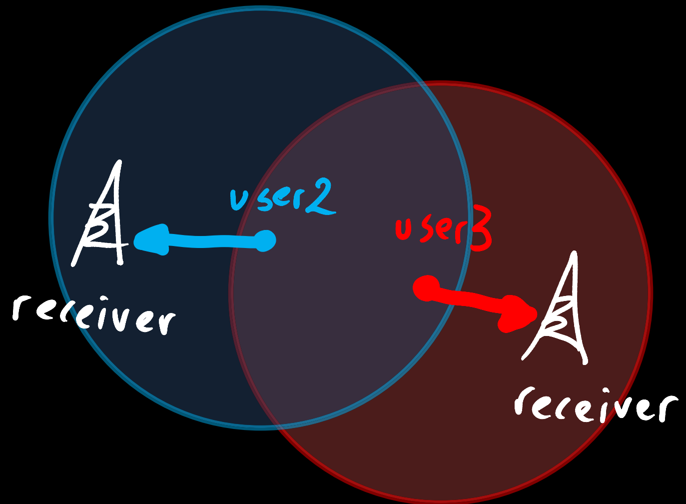
MAC protocols typically involve a random wait before transmission, and longer random back-off in the event of a collision.



In a network setting, it is more challenging to devise a good MAC protocol.



The hidden terminal problem. User 1 does not hear users 2 or 3, so it does not know to back off, so the receiver is overloaded.



The exposed terminal problem. Each user backs off when the other is transmitting, but this is not necessary since the receivers do not hear the collision.

What are the design goals of a MAC protocol?

Let θ_i be the rate of successfully delivered packets for user *i*. Let user *i* have a utility function $U_i(\theta_i)$. A reasonable objective is

$$\max \sum_{\text{all users } i} U_i(\theta_i)$$

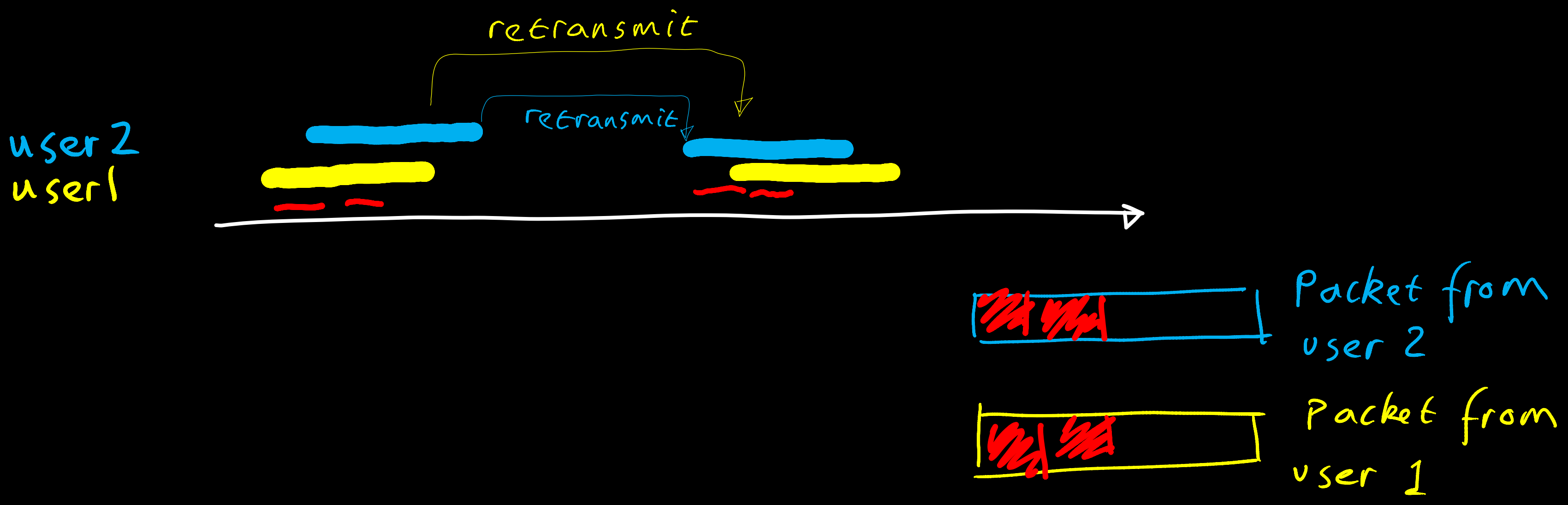
$$\text{over attainable } (\theta_i,\ 1 \le i \le n)$$

The MAC layer has to discover how many users there are, and what rates are attainable, and solve this optimization.

It should be a distributed algorithm, and it should react quickly when the network changes. It should work in a multihop network.

A remarkable recent MAC algorithm shows a new way of thinking about contention

"*ZigZag decoding: combating hidden terminals in wireless networks*", Gollakota and Katabi, SIGCOMM 2008



- The wireless medium is additive. Therefore, even when transmissions collide, ZigZag may be able to decode the signal.
- But this decoding only works when there are few packets colliding. ZigZag relies on ALOHA-style adaptive backoff, to ensure that collisions are rare.

I propose a toy model for MAC, with which to explore some new styles of resource allocation

- Let there be *n* users, each with an infinite backlog of packets. Over a long timescale, *n* varies.
- Consider a slotted-time binary noiseless additive channel.

Let $x_{it} \in \{0,1\}^K$ be the signal transmitted by user i in timeslot t.

(No transmission: $x_{it} = 0$.)

Let $Y_t = \sum_i X_{it} \pmod 2$ be what the receiver hears in timeslot t.

- How should users choose what to send, in order to maximize aggregate utility?

A simple MAC encoding

Suppose there are two users. Let user 1 have packets $P_1, P_2, \ldots \in \{0,1\}^K$ to send. Let user 2 have packets $Q_1, Q_2, \ldots$

Let $\mathcal{A} \subset \{0,1\}^{K \times K}$ be a set of invertible matrices. Let $A_1, A_2, \ldots$ and $B_1, B_2, \ldots$ be sequences of IID matrices drawn from $\mathcal{A}$.

User 1 sends	$A_1 P_1$	$A_2 P_1$	$A_3 P_1$	$A_4 P_2$	$A_5 P_2$	$A_6 P_2$	$A_7 P_3$
User 2 sends	$B_1 Q_1$	$B_2 Q_1$	$B_3 Q_2$	$B_4 Q_2$	$B_5 Q_3$	$B_6 Q_3$	$B_7 Q_4$
Receiver hears	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7

A simple MAC decoder

Suppose the receiver knows

- the sequences $A_1, A_2, \ldots$ and $B_1, B_2, \ldots$
- which packets are transmitted in which timeslots

The receiver also knows,

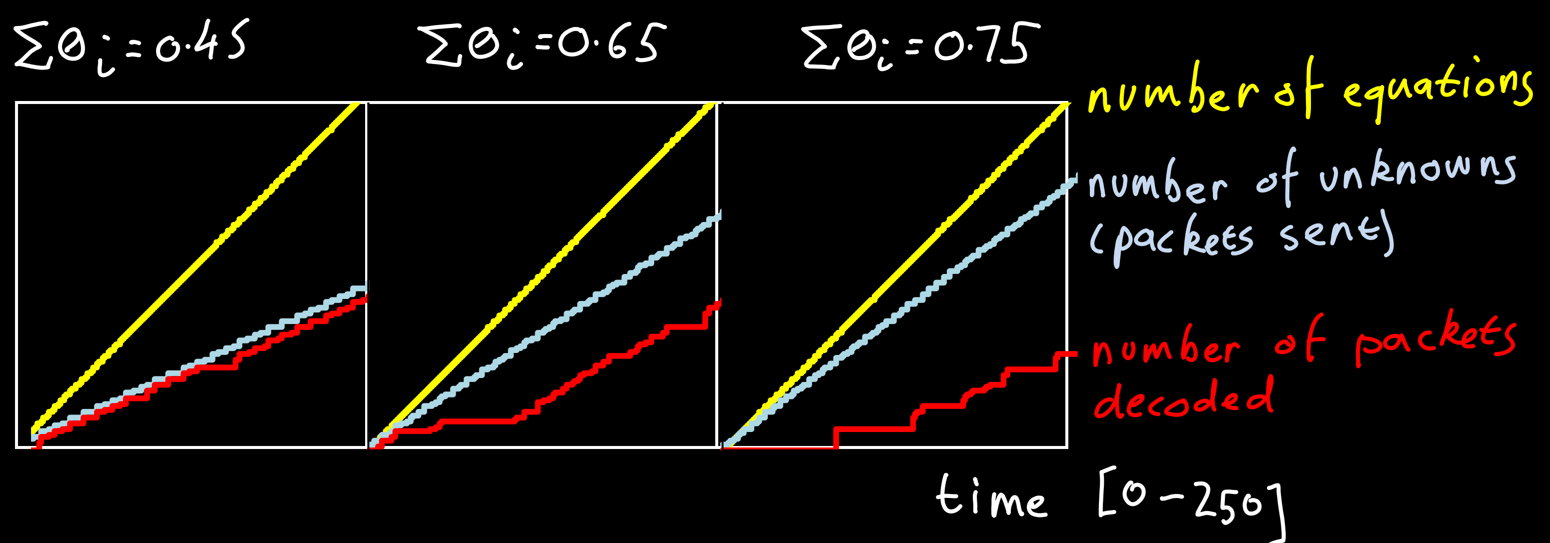
		# eqns	# unknowns
at time 1, that	$A_1 P_1 + B_1 Q_1 = Y_1$	1	2
at time 2, that	$A_2 P_1 + B_2 Q_1 = Y_2$	2	2
at time 3, that	$A_3 P_1 + B_3 Q_2 = Y_3$	3	3
at time 4, that	$A_4 P_2 + B_4 Q_2 = Y_4$	4	4
…		⋮	

How well does this MAC decoder work?

- Let user *i* introduce new packets at rate θ_i
- The receiver acquires new equations at rate 1 / timeslot
- New unknowns are introduced at rate $\sum \theta_i$ / timeslot

- This decoder's success rate depends somehow on $\sum \theta_i$ and on the degree of linear independence in the equations.

Simulation of 4 sources, $K=15$



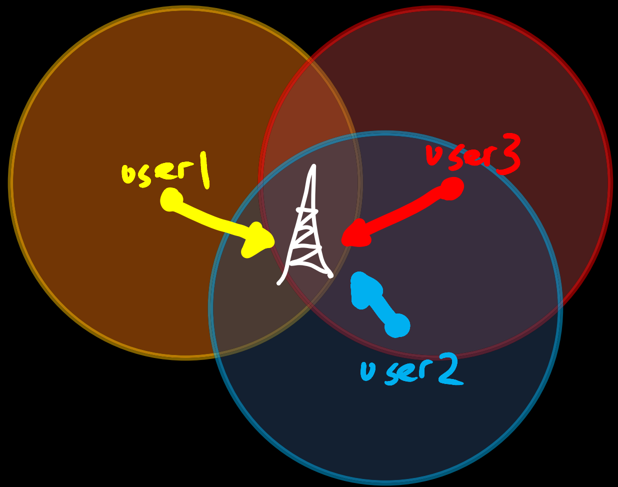
A simple congestion controller

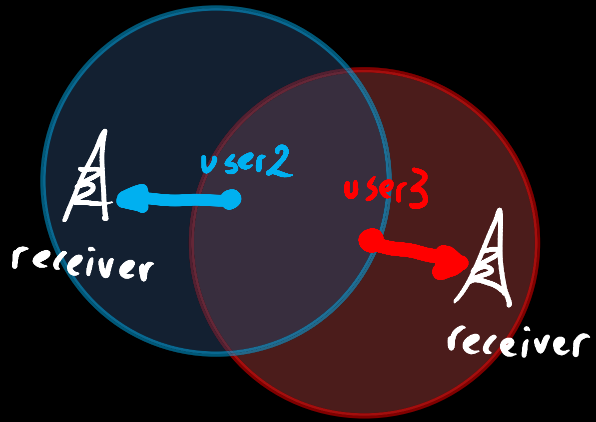
When the receiver fails to decode some packets, let it send a congestion notification to all the users whose signals it hears. (This notification might be "out of band".)

Let each user i adapt θ_i according to his utility function

- increase θ_i steadily when there is no congestion
- decrease θ_i when it receives congestion feedback.

This should automatically solve the network utility maximization problem.





MAC3
medium access
Coding & Congestion Control

The wireless medium is additive.

ALOHA-style back-off protocols are a type of coding. There are other types of coding that take advantage of additivity: for these codes it is useful for users to keep sending all the time, so that the channel is never wasted.

It is necessary to study MAC design and channel coding together.

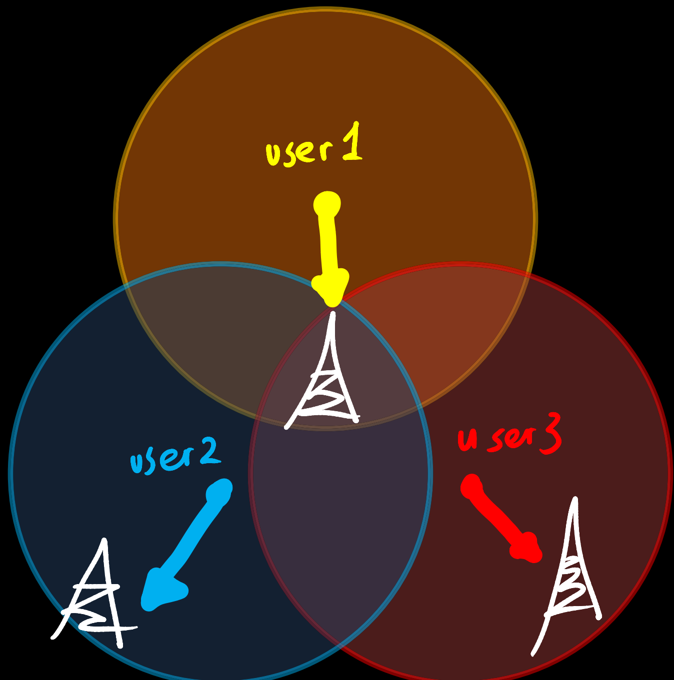
It is possible to combine coding with utility-maximizing congestion control. This automatically solves the hidden terminal and exposed terminal problems, and it applies automatically to multihop networks.

Some questions

- For the MAC I described, what is the probability of successfully decoding a packet? How does it depend on $\mathcal{A}$ and on $(\theta_i, 1 \leq i \leq n)$?

- What is a better MAC?

- How can the good features of this toy model, especially the congestion control mechanism, be implemented in a practical system?



Our naïve decoder requires the receiver to be able to decode everything it hears, giving the capacity bound

$$\theta_1 + \theta_2 + \theta_3 \leq 1.$$

Shannon's bound suggests we should be able to do better:

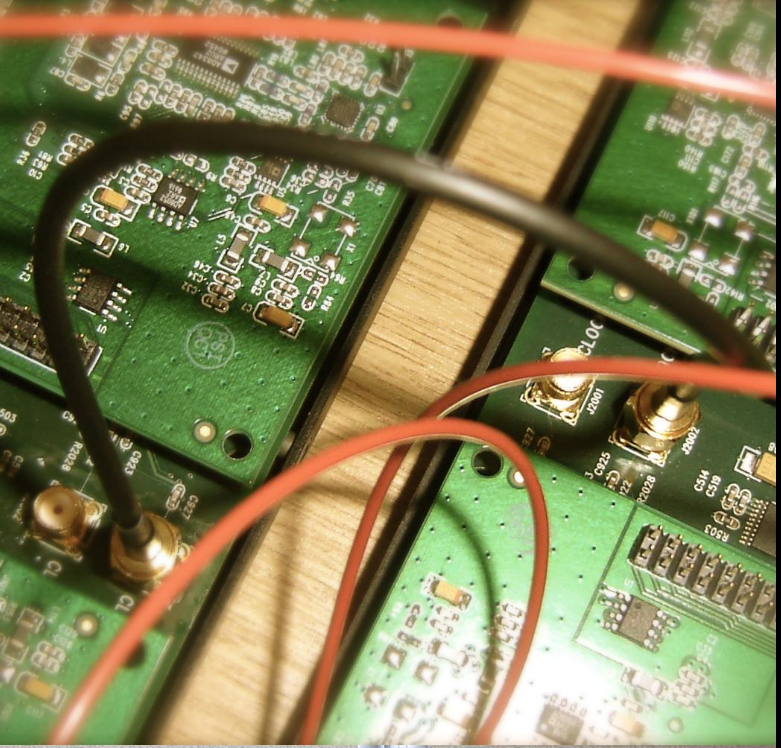
$$\theta_1 + \mathrm{H}\left(\mathrm{H}^{-1}(\theta_2) \oplus \mathrm{H}^{-1}(\theta_3)\right) \leq 1$$

where $p \oplus q = p(1-q) + q(1-p)$. How might we achieve this?

If the three users coordinate their transmissions, the capacity constraint is even better:

$$\theta_1 + \max(\theta_2, \theta_3) \leq 1.$$

How might users coordinate themselves?



Software radios invite us to rethink how networks are organized – to rethink the layers of the network stack.