MIMO As a First-Class Citizen in 802.11

Kate C.-J. Lin
Academia Sinica

Shyamnath Gollakota and Dina Katabi
MIT

present MIMO as a first-class citizen in 802.11.
The traditional 802.11 protocol was designed for single-antenna systems.
When MIMO emerges, it’s just an add-on.
However, if we rethink 802.11 design for MIMO, we can get a much higher throughput.
MIMO technologies are emerging, and we are having more and more antennas in a single device.

But, on the other hand, there are still many devices limited by their physical size, so they can only support a single antenna. There are also some in the middle. So, what we see today is that wireless devices increasingly have heterogeneous numbers of antennas and different mimo capabilities.
However, 802.11 was originally designed for single-antenna devices.

In today’s 802.11, when a single-antenna node transmits, all mimo nodes refrain from transmitting.
This is however undesirable because, in principle, mimo nodes should be able to receive multiple packets concurrently.

For example, Bob’s receiver has two antennas, so should be able to receive two concurrent packets.
Given that Alice is only transmitting one packet on the medium, Bob can actually transmit another packet to his receiver.
Similarly, Chris’s receiver has three antennas, so, supposedly, can get another packet from Chris, in the presence of two ongoing transmissions.

Ok, great, we have concurrent transmissions.
But how about Alice’s receiver? Alice’s receiver only has one antenna, so he can only decode one packet. If Bob just transmits, he is gonna interfere with Alice’s receiver.

Similarly, if Chris just transmits, both Alice’s and Bob’s receivers will get hurt.

So, the question is how can we transmit concurrently, but without interfering with any existing transmissions?
Our goal is to have concurrent transmissions across mimo nodes, but without harming all ongoing transmissions.

In this talk, I’m going to introduce 802.11n+.
N+ is the first protocol that allows MIMO nodes to join ongoing transmissions without interfering with them.

Not only this, but also it maintains all the great properties of random access, which is the main reason why we love today’s 802.11 because it allows us to access the medium in a fully distributed way.

Finally, n+ is implemented, and shown in practice to significantly improve the throughput.
To achieve our goal, we will need to answer two questions.

1. How to transmit without interfering with ongoing transmissions?

2. How do we achieve it in a random access manner?

To achieve our goal, we will need to answer two questions.

How to transmit concurrently?

And how to do it in random access?
1. How to transmit without interfering with ongoing transmissions?

2. How do we achieve it in a random access manner?

Let’s start by answering our first question. How can we transmit without introducing interference?
The answer to that is interference nulling.
So, how does it work?
Say Alice is already transmitting a packet.
And Bob wants to transmit a concurrent packet $y$ to his receiver.

To ensure not to interfere with Alice’s receiver, Bob can null his signal at Alice’s receiver.
In particular, he can use his first antenna to transmit his packet $y$ multiplied by some given variable $\alpha$.
The signal goes through a channel $h_1$.
Then, Alice’s receiver will see an interference $h_1 ay$ from Bob.

Bob can use the other antenna to transmit by
Then, the signal goes through a different channel $h_2$.
So, now Alice’s receiver will get an interference, which is the summation of these two signals from Bob.

Nulling the signal means that we want to force this interference to be 0, which is doable. Regardless of the channel values, we can always pick some alpha beta to satisfy this constraint such that two signals will sum up to 0 at Alice’s receiver.
Then, Alice’s receiver will not see any signal from Bob.
So, this is how nulling works.

But, one thing worth to notice is that such nulling doesn’t prevent Bob’s receiver from receiving his signal.
This is because the channels received by Bob’s receiver are different from $h_1$ and $h_2$.
Cancelling signals for Alice’s receiver doesn’t mean that cancelling signals for Bob’s receiver.
That’s why Bob can transmit a packet to his receiver.
So far, we have an answer to our question. To transmit without interfering with ongoing transmission, we can use nulling and now we know how to null the signals.

But are we done?
What if we have more mimo nodes, like Chris, who also wants to transmit concurrently
Can we just use nulling to allow Chris to transmit without interfering with either Alice’s receiver or Bob’s receiver?

Unfortunately, the answer is no.

Turns out that when we want to support more concurrent transmissions, nulling alone is not enough.

For example, if Chris wants to transmit concurrently, he should not interfere with both Alice’s receiver and Bob’s receiver.
To do that, Chris will need to cancel his signals for three antennas, one at Alice’s receiver and two at Bob’s receiver.

The bad news is it is not doable.

This is because a 3-antenna transmitter cannot null his signals at 3 receive antennas.

Let’s see why.
If Chris wants to transmit a packet $z$, as before, he can transmit a weight version of the signal $z$.

We can write the nulling equations.
We will have one equation for each antenna we want to null the signal, which should be set to 0.

We don’t need to worry about these equations, because the details are not so important.
The important thing is that the only solution for these equation is to set all variables to 0, which means that Chris is not transmitting at all.
Clearly, this solution is clearly not what we want, because this means that Chris cannot transmit anything to his receiver.
This is why nulling doesn’t work.

But, are we doomed? Is there any way to get three concurrent transmissions?

The reason why nulling is not doable is because making your signal invisible is a very strong constraint.

But, does Chris really need to make his signals invisible for every receiver?

Fortunately, no!
I’m going to show you that Chris doesn’t really need to cancel his signals for both Alice’s receiver and Bob’s receiver.
In fact, Chris can use a new technique, called interference alignment, to transmit his packet.

Let me explain to you how interference alignment works.
To understand what is alignment, I will need to introduce to you some very important properties of MIMO.

The first thing we need to know is that an n-antenna node receives in n-dimensional space. What does that mean?

If I have a single-antenna receiver, I receive signals at only one antenna, so I receive signals in one-dimension.

But, if I have two antennas, I will receive the signal at the first antennas, and also receive the signal at the second antenna. So really I’m receiving a vector in a two-dimensional space.

Similarly, a three-antenna node receives at each antenna a different signal, so the received signal is a vector in a three-dimensional space.

We can even generalize it to an n-antenna node, which will receive in n-dimensional space.
The second thing we need to know is that the transmitter can rotate the received signal to whatever direction he wants.

For example, if someone is transmitting a signal, then this 2-antenna receiver receives a vector \( y \) in his two-dimensional space. Say we want to rotate the received signal to \( y' \).

We can easily do that.

First thing to note is that we can always write \( y' \) as \( Ry \), where \( R \) is a rotation matrix.

So, to rotate the received signal \( y \) to \( y' \), all the transmitter has to do is to multiply its transmitted signal by the same rotation matrix \( R \).

So, now we can rotate the signal to a new direction \( y' \).
With this basic understanding about mimo, next, let me use a very simple example to explain what is interference alignment.

Let’s consider a 2-antenna node who is receiving one signal he wants and another signal he doesn’t want.

Because he has two antennas, he can decode these two signals to get the signal he wants.

But, say that we have the second interferer.
Now this two-antenna receiver cannot decode these three signals in his two-dimensional space.
So, he cannot get the signal he wants.

But if this new interferer can somehow align his signal along the direction of the first interferer by rotating his signal,
Interference Alignment

2-antenna receiver

N-antenna node can only decode N signals

If \( I_1 \) and \( I_2 \) are aligned,
\[ \rightarrow \text{appear as one interferer} \]
\[ \rightarrow \text{2-antenna receiver can decode the wanted signal} \]

, this receiver will see only one interference.
Then, it looks like there are only two signals on the medium, so he can still decode two signals.
One is the signal he wants, the other is the sum of two interferences he doesn’t want.

Ok, now we know how to align the signal.
Ok, let’s go back to see why this alignment is a very nice property for solving our problem.

Remember that Chris doesn’t want to interfere with both Alice’s receiver and Bob’s receiver.

For Alice’s receiver, Chris can still use nulling to cancel his signal. That’s fine.

However, for Bob’s receiver, we are going to use this new concept: interference alignment.

What we do is to leverage the fact that Bob’s receiver is already receiving some signals from Alice,
But he doesn’t want that signal.
So it doesn’t hurt to put another interference on top of Alice’s signal.

Basically, Chris doesn’t really need to completely cancel his signal for Bob’s receiver.
All he has to do is to align his signal with Alice’s signal.
As a result, from the perspective of Bob’s receiver, it looks like there are still only two signals, one is Bob’s signal, the other is this garbage signal, which is the combination of Alice and Chris.
So, he can still decode these two signals.

Even though he cannot decode Alice’s and Chris’s signal individually, but it doesn’t matter because he doesn’t want these two signals.
The important thing is that he can get what he wants, which is Bob’s signal.

Ok, we’ve shown that we can use nulling and alignment to transmit three concurrent packets.
Therefore, n+ allows all senders to transmit, but the throughput will be as high as if only the three-antenna sender is transmitting all the time.
And this is the maximum throughput we can get for this topology.

Of course, there are more details about how to fully implement interference nulling and interference alignment, and these are described in the paper.
The thing I want to emphasize is that we can do the same process for any generic scenario.
Specifically, in our paper,
we have a general protocol, which allows each sender to compute where and how to null, also where and how to align in a distributed way.

We also proved that the number of concurrent streams can be equal to the maximum number of antennas supported by any sender.
1. How to transmit without interfering with ongoing transmissions?
   - Interference nulling
   - Interference alignment

2. How do we achieve it in a random access manner?

So, we achieve our goal. Now, we know how to use the combination of nulling and alignment of transmit concurrently.
1. How to transmit without interfering with ongoing transmissions?
   - Interference nulling
   - Interference alignment

2. How do we achieve it in a random access manner?

But, the second question is can we achieve this goal without loosing random access?
Even though I know how to use nulling and alignment to transmit concurrently, but how can I know who is transmitting, and how many transmissions are already on the medium.

To get this information, one possible solution is to have all nodes send the information to a central controller.
Then the central controller can do some scheduling and tell Bob and Chris, say, hey you guys both can transmit a packet concurrently with Alice..

However, this is not what we want because we are going to lose everything we love about 802.11, which is random access, which is no coordination, which is no scheduling.

What we really want is to be able to transmit concurrently, but, at the same time, maintain the great random access nature.
In 802.11, random access works because it allows us to use carrier sense to contend for the medium in a fully distributed way.

However, carrier sense doesn’t work for concurrent transmissions. Because we cannot just check whether the medium is idle or not. Otherwise, we are not going to have concurrent transmissions across different nodes.

So, what we need is to extend carrier sense to work despite ongoing transmissions. But how can we do that? Our solution is multidimensional carrier sense.
Consider a two-antenna node Ben who is interested in concurrent transmissions.

Ben needs to distinguish between the following two scenarios.
There is only one transmission on the medium, or
There are two transmissions on the medium.

Because if there is only one transmission on the medium, he has two antennas, so he can sneak out another packet.
But if there are already two transmissions, he should not transmit.

As I said earlier, because Ben has two antennas, he receives in a two-dimensional space.
So, what he wants is to distinguish between this scenario where there is only one vector from Alice’s transmission, and the second scenario where there are two vectors from Alice and Bob.

This is a simple linear algebra problem that we can solve using say Gaussian elimination.

So, let’s see how can we use this concept in the protocol.
Let’s use the same example where Alice is transmitting. All the nodes that have more than 1 antenna, like Bob and Ben, transmit concurrently with Alice. So, they need to contend with each other, but how can they contend in the presence of Alice’s signal?

As I said, Bob and Ben both have two antennas, so they both see Alice’s signal in a two-dimensional space.
Given this mimo property, the basic idea of our multi-dimensional carrier sense is that Bob and Ben can both project orthogonal to Alice’s signal.
In particular, they can both project the signal along a direction orthogonal to Alice’s signal. After projection, they will not see Alice’s signal, so the medium looks idle.
Therefore, they can simply apply standard 802.11 carrier sense.
They both listen to the medium after projection.

One of them will win the contention and transmit first, say Bob.

Now, Ben starts seeing some signals from Bob. Then, after projection, he will detect energy in this orthogonal direction, and can identify that there are two transmissions now.

Because he only has two antennas, he knows that he lost the contention, and should not transmit.

Effectively, we’ve extended carrier sense to work despite ongoing transmissions.

Here we use two-antenna nodes to explain our protocol, but everything works for arbitrary number of antennas.
1. How to transmit without interfering with ongoing transmissions?
   - Interference nulling
   - Interference alignment

2. How do we achieve it in a random access manner?
   - Multi-dimensional carrier sense

So what we have now is a protocol, which allows us to transmit concurrently, but at the same time, maintain the random access nature by using multi-dimensional carrier sense.
Ok, let’s see how n+ performs.
We implement n+ in USRP.

Our implementation is based on OFDM, and uses 802.11 modulations and convolutional codes.
For each experiment, we randomly pick a subset of locations and put the nodes over there.
1. How to transmit without interfering with ongoing transmissions?
   - Interference nulling
   - Interference alignment

2. How do we achieve it in a random access manner?
   - Multi-dimensional carrier sense

We are going to check the performance of each component of n+.
Let’s first check whether we can null the signals correctly.

1. How to transmit without interfering with ongoing transmissions?
   - Interference nulling
   - Interference alignment

2. How do we achieve it in a random access manner?
   - Multi-dimensional carrier sense
In this experiment, Alice is sending a signal to her receiver, and Bob then transmits concurrently.

We want to check whether Bob can in practice null his signal at Alice’s receiver.
On the x-axis, we will look at the SNR of the unwanted signal before nulling.
On the y-axis, we are going to look at the residual interference from Bob at Alice’s receiver.
We measure it in dB.
Ideally, we want this value to be 0.

We will plot the results up to 25 dB, which is the SNR of interest in 802.11
Here are the results.

The higher SNR of Bob’s signal, we can see a higher nulling error. This is because it is harder for Bob to cancel a higher power signal.

However, in all cases, the residual interference is small, which is at most about 1db.
We did the similar experiment for alignment.
Let me show you the residual interference after alignment.

As we can see, the error for alignment is higher than nulling.

This is because two interferences are from two independent senders, Alice and Chris.
So, it is harder to align two independent signals accurately.

However, even though alignment is harder, the residual interference after alignment is fairly small, which is less than 1.5dB.
Finally let’s check whether we can really do carrier sensing in the presence of ongoing transmission.

1. How to transmit without interfering with ongoing transmissions?
   ▶ Interference nulling
   ▶ Interference alignment

2. How do we achieve it in a random access manner?
   ▶ Multi-dimensional carrier sense
We next want to check whether we can do carrier sense despite ongoing transmissions. In our experiment, we have a three-antenna node how is performing carrier sense.

We compare the difference between traditional CS and carrier sense after projection.

Each graph plots the power sensed on the medium as a function of time.

Each experiment has two phases. There is only one transmissions in phase, and the second transmission joins in phase 2. We focus on a challenging scenario, where the 2\textsuperscript{nd} transmission has a significantly weaker power than the 1\textsuperscript{st} one. We want to check whether we can still identify the second transmission even if it has a very low power.
Here is the result for traditional CS.
As we can see, the sensed signal strength doesn’t increase much in phase 2, so it is really hard to identify whether the second transmission has already joined.
Here is the result after projection.

Because we project orthogonal to ongoing transmissions, we can see a big 9dB jump when the second transmission starts in phase 2.

So, we can easily identify there are two concurrent transmissions on the medium.
We saw that the error from nulling and alignment is less than 1.5dB, but what we are really interested in is whether n+ can deliver throughput gain despite these errors.

So, let’s check the total throughput using this topology. We compare 802.11 with our n+. 
The x-axis is the total throughput of three receivers, and the y-axis is the CDFs taken over all experiments.
Here are the results for 802.11

Each point on the CDF is the throughput of one experiment. We get different points by repeating the experiment for different node locations in the testbed.
Here are results for n+.

As you can see, even for this small topology, n+ significantly improves the throughput.
In particular, the median throughput increases by about double.
Therefore, we see that n+ can deliver a significant throughput gain in practice.

Here, I show you the results for this topology.
But, in the paper, we have results for more sophisticated topologies.
Our work builds on past works on information theory about signal precoding and multi-user mimo. We also build on previous practical mimo systems, including Beamforming, SAM, and IAC.

However, our work is the first random access protocol that allows us to have concurrent transmissions across mimo nodes, but without any central coordination.
To conclude,

In today’s 802.11, MIMO is only an add-on, but in n+ MIMO is a first-class citizen.

N+ is the first protocol which allows us to have more concurrent transmissions, but at the same time we don’t need to give up random access.

We empirically show it is practical.