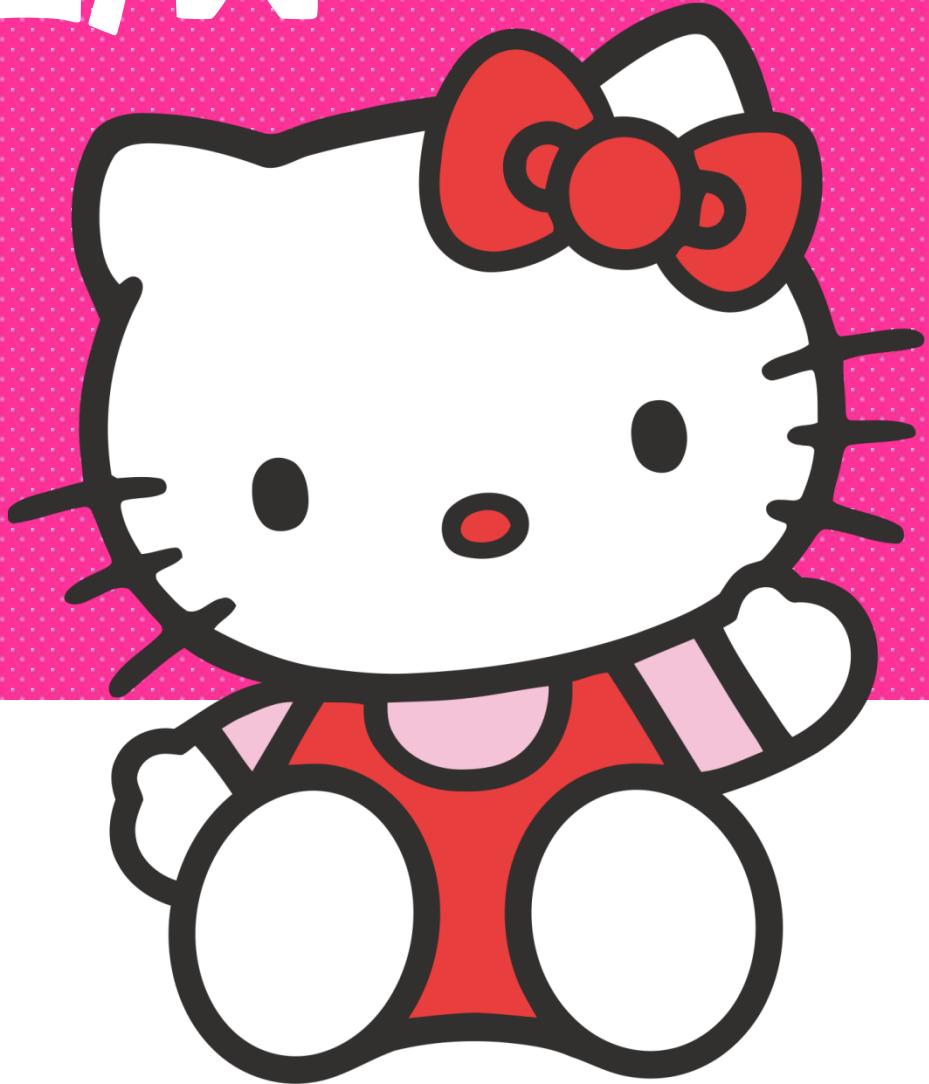


DVB-S2/X

Block Party!



**Hosted by
Phase 4 Ground**



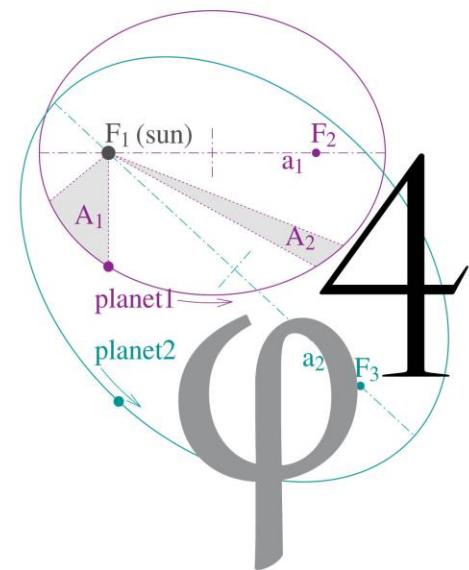


Who are you?

- W5NYV
- MSEE Information Theory from USC
- Active in IEEE Information Theory Society
- Co-founder of Open Research Institute
- Lead of Phase 4 Ground
- DVB-S2 and DVB-S2X are central technologies
- The receivers need to be in GNU Radio
- The hardest part is considered to be LDPC decode

Who are we?

φ^4



Open Research Institute, Inc. (ORI) is a non-profit research and development organization which provides all of its work to the general public under the principles of Open Source and Open Access to Research. openresearch.institute

Phase 4 radios

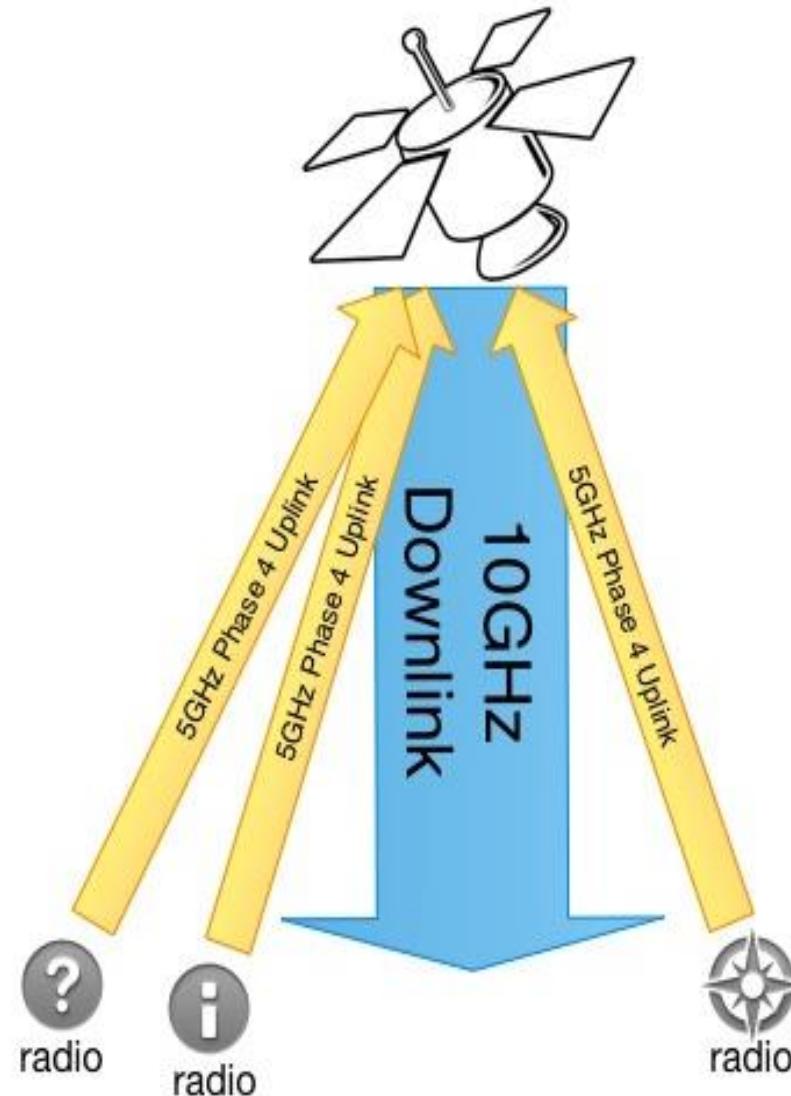
5.645-5.655 GHz up

10.45-10.46 GHz down

MSK channelized uplink

DVB-S2/X downlink

Amateur space and
terrestrial sub-bands
are right next to each other.



What is the team doing?

RTP multicast innovations (Phil Karn)

DVB correlator (expand the one in GR)

Open Source LDPC decode, GPU to FPGA

Dual Band Feed, 5GHz and 10GHz

GSE published work (wireshark, GR)

Filters! 5GHz amps!

ARAP demonstrations*

Having tons of fun

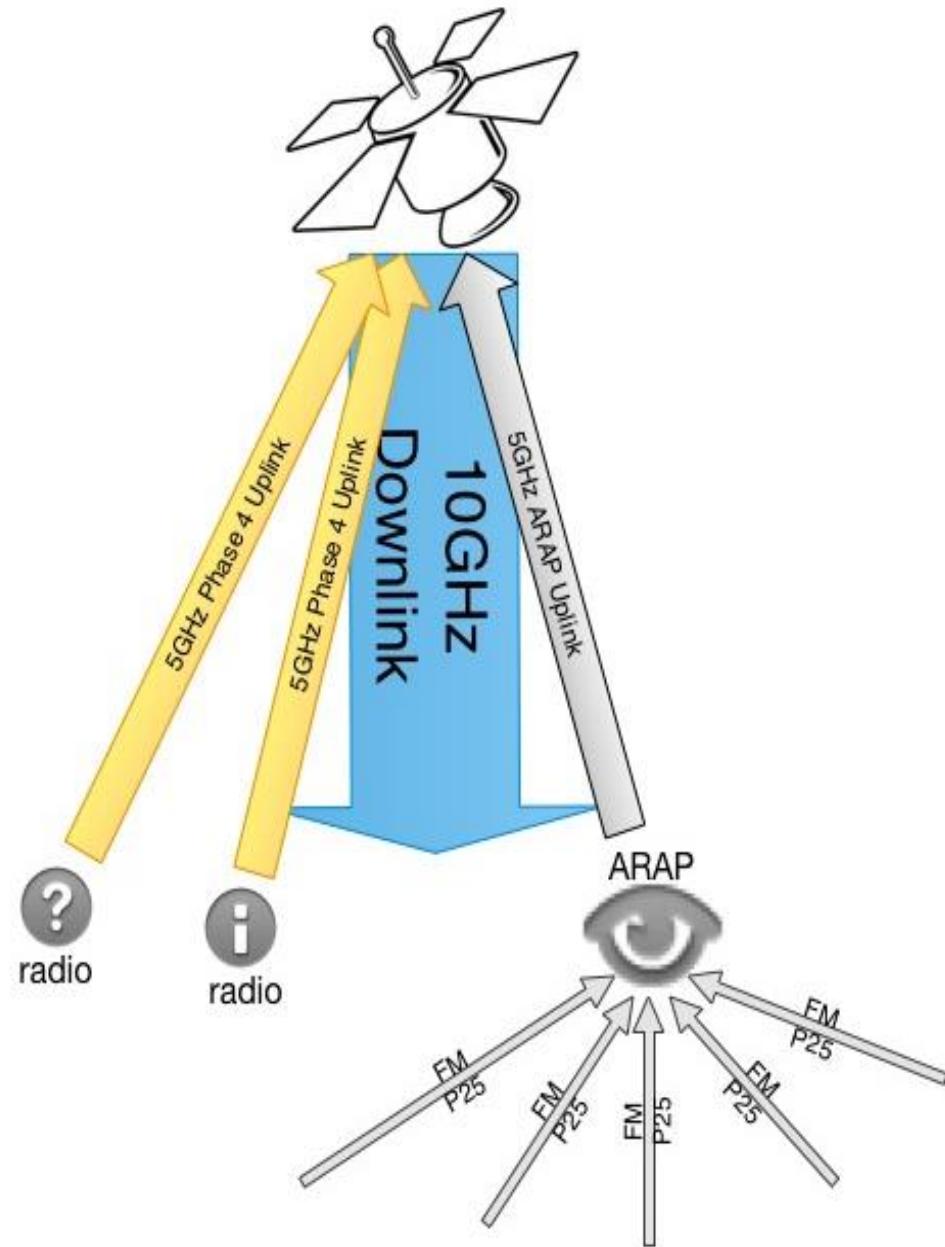
Buying every SDR dev board we can find

Throwing a Block Party!

Phase 4 radios

5.645-5.655 GHz up

10.45-10.46 GHz down





LDPC - A Compelling History

- Discovered in 1963 by Robert Gallagher.
- Impractical to implement. Promptly “lost”.
- Interpreted in graphical form by Michael Tanner in 1981. We use Tanner Graphs today.
- Still impractical to implement. “Lost” again.
- Revived by Sir David MacKay in 1996 in a paper.
- Chosen for DVB-S2 standard in 2003.
- Since then, LDPC has surpassed Turbo codes in error floor and performance, especially at higher code rates. Huge area of academic research in FEC.



How Good are LDPC codes?

- “Long LDPC codes with iterative decoding based on belief propagation are a fraction of a dB from the Shannon limit.” Implementable decoders!
- LDPC codes don’t need interleavers at all or trellis on decode. Parallelizable decoders!
- They have great block error and burst error performance.
- Error floors occur at lower bit error rates.
- Decoding complexity increases linearly with the number of entries (number of 1s) in the parity check matrix. Keep it low!
- Single code good for many channel models.



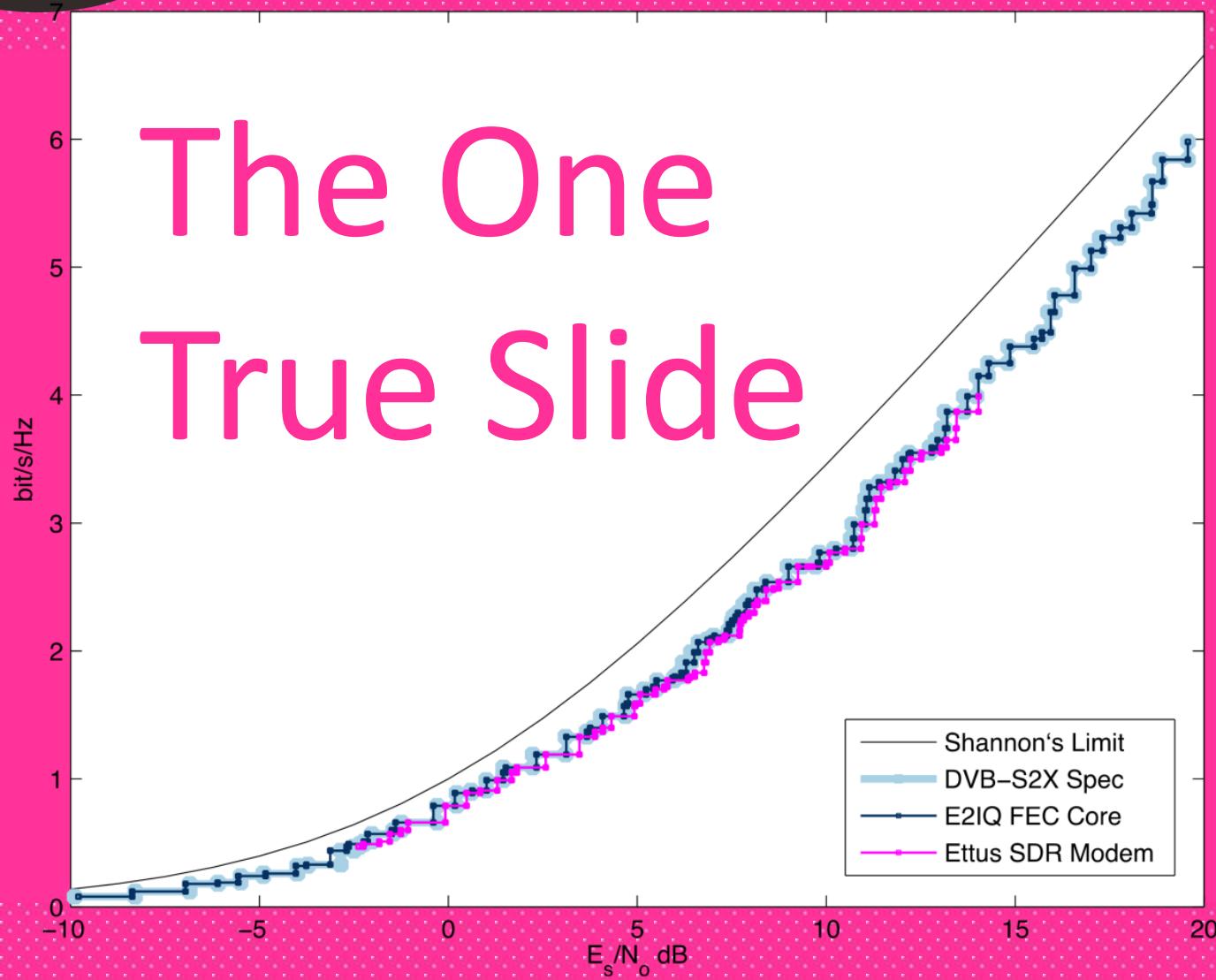
In Phase 4 Ground ACM!

- DVB-S2/X has something called ACM.
- Adaptive Coding and Modulation lets you adjust to changing signal-to-noise ratios on a per-frame basis.
- Best possible throughput for the channel conditions with the radio you have.

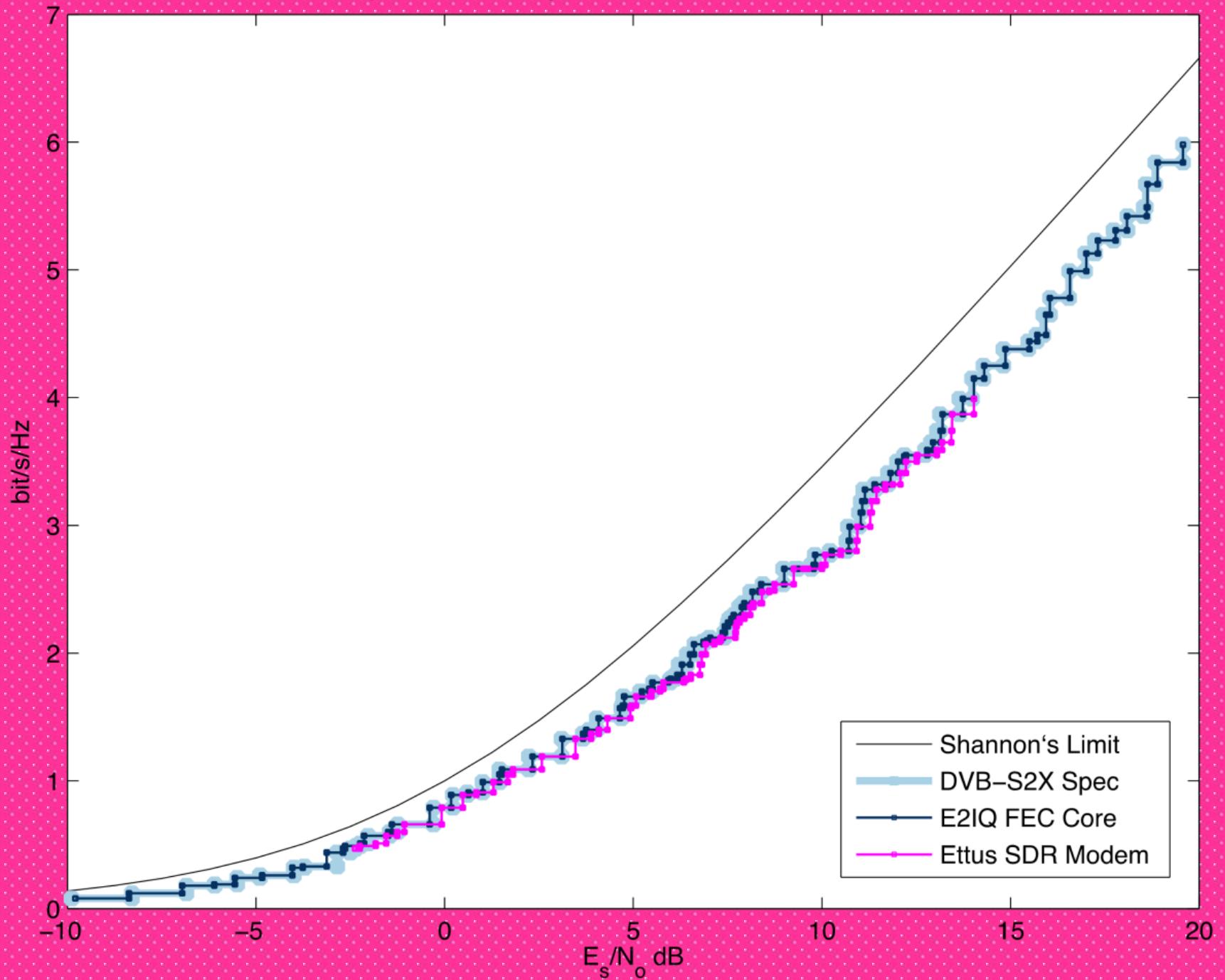


Why Use ACM?

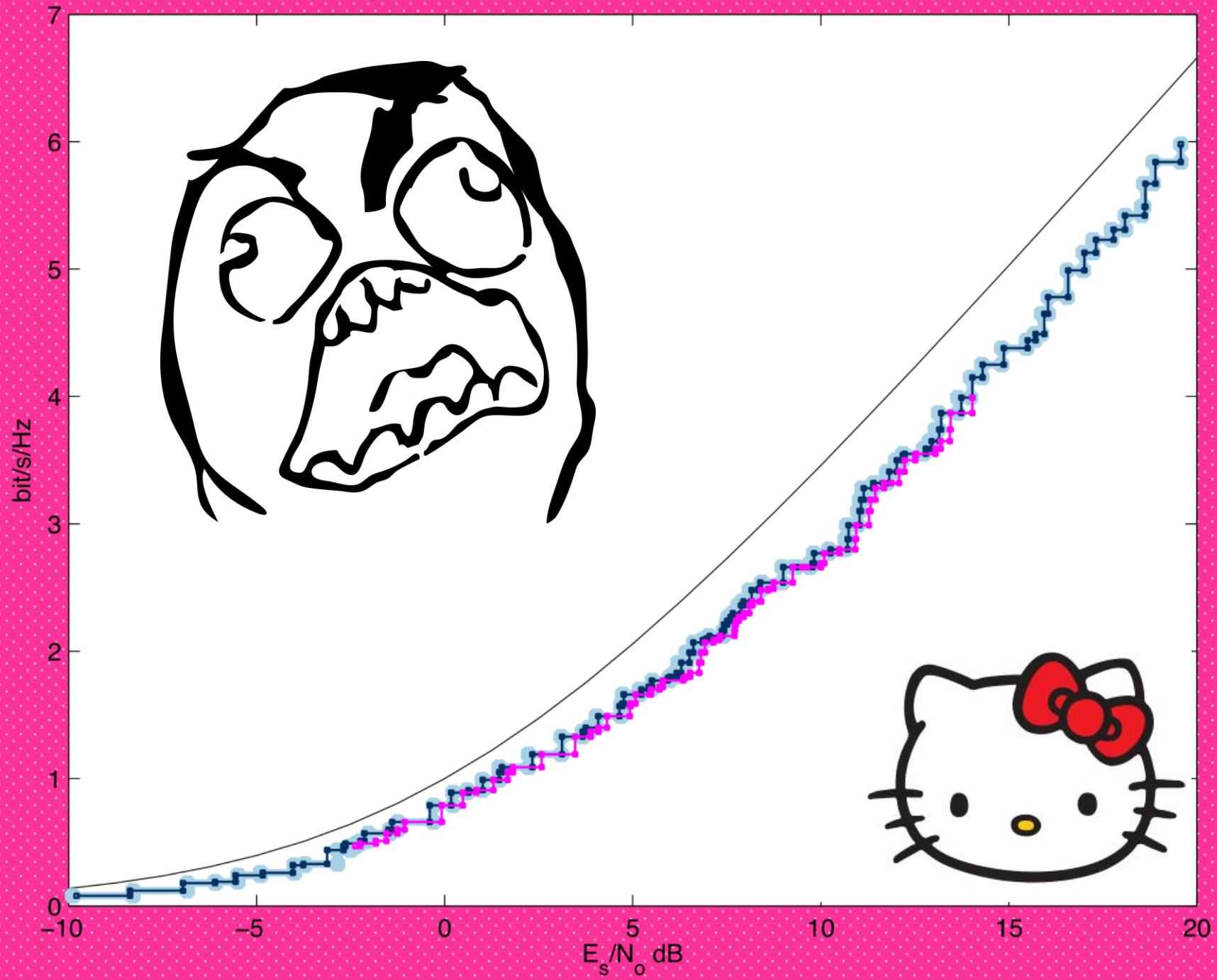
Spectral Efficiency of E2IQ FEC Core and Modem



Spectral Efficiency of E2IQ FEC Core and Modem

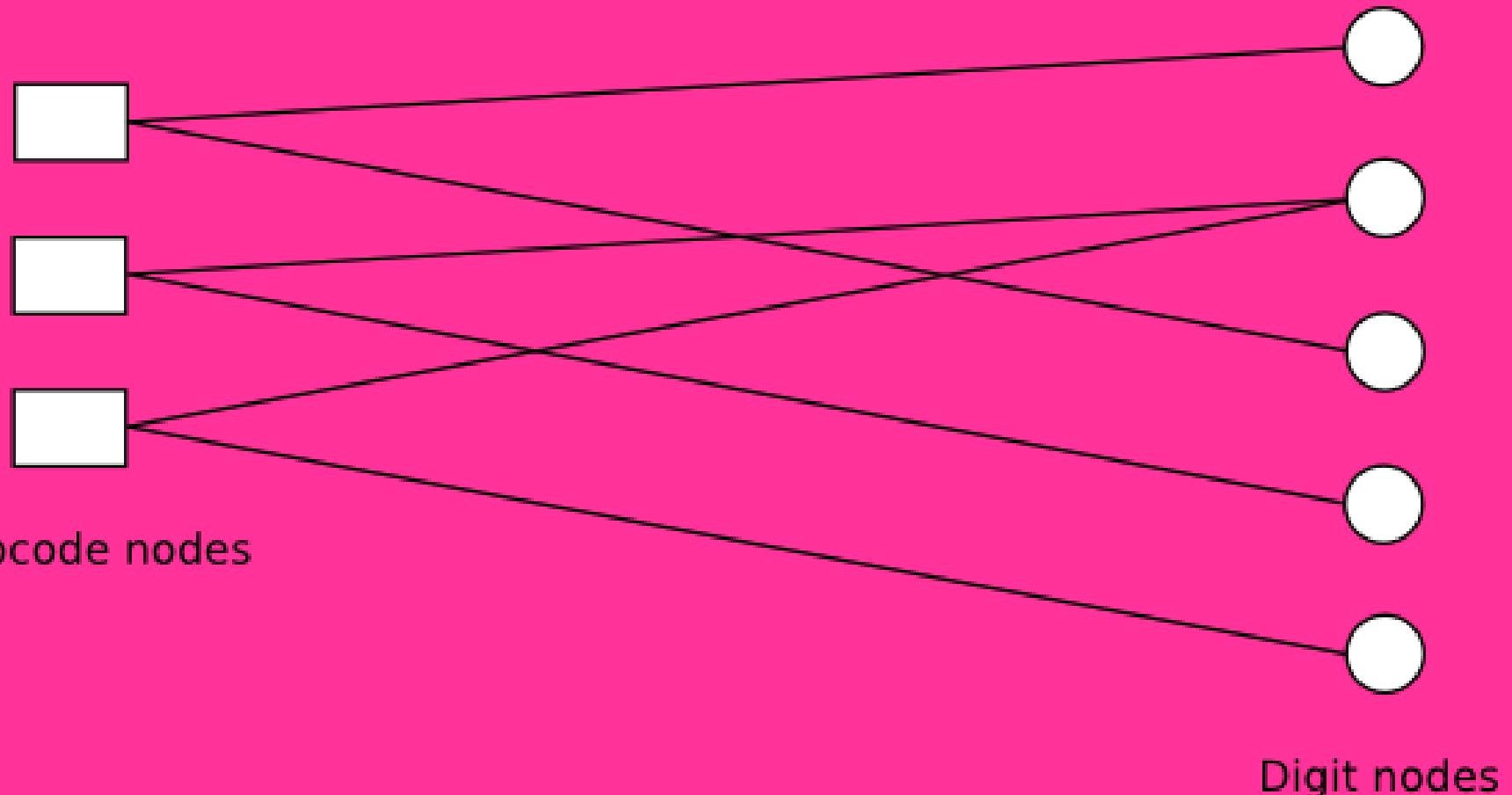


Spectral Efficiency of E2IQ FEC Core and Modem





(Michael) Tanner Graphs!



$H =$

1	1	1	1	0	0	0	0	0	0
1	0	0	0	1	1	1	0	0	0
0	1	0	0	1	0	0	1	1	0
0	0	1	0	0	1	0	1	0	1
0	0	0	1	0	0	1	0	1	1

J rows = J check nodes (J parity check sums)

n columns = n variable nodes (n bits in a codeword)

S1

S2

S3

S4

S5

v1

v2

v3

v4

v5

v6

v7

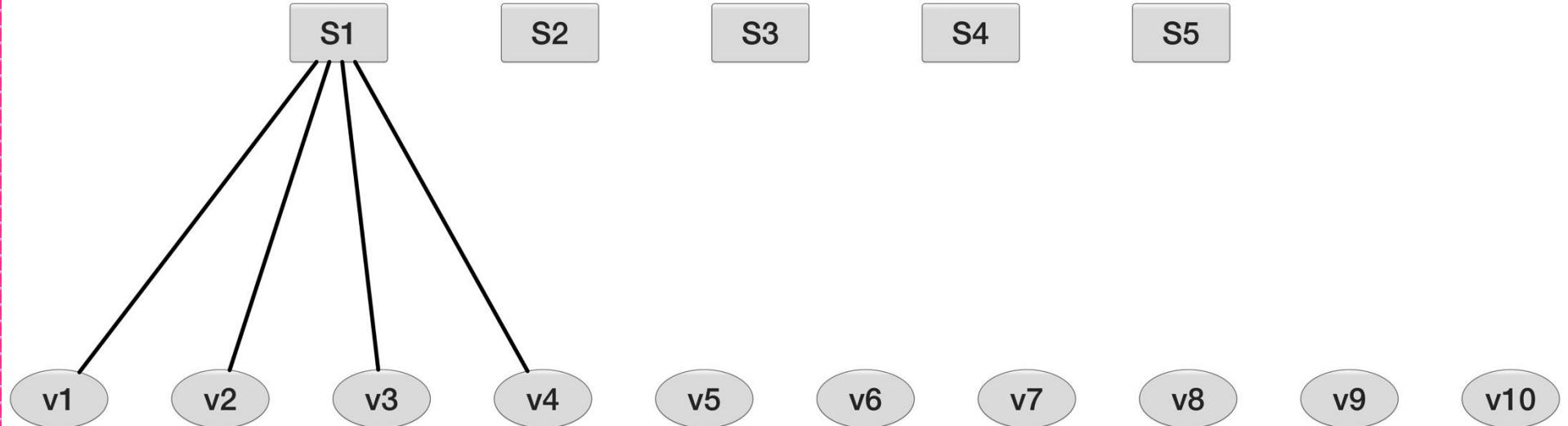
v8

v9

v10

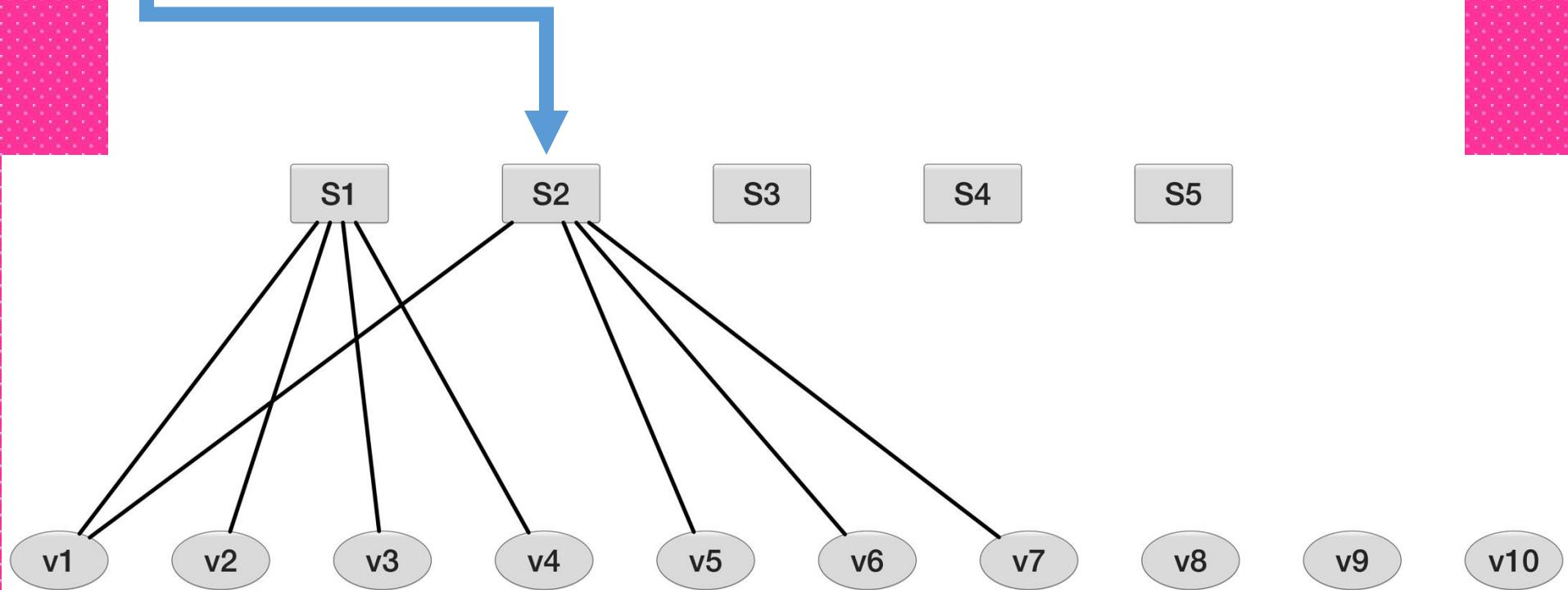
$H =$

1	1	1	1	0	0	0	0	0	0	0
1	0	0	0	1	1	1	0	0	0	0
0	1	0	0	1	0	0	1	1	0	0
0	0	1	0	0	1	0	1	0	1	1
0	0	0	1	0	0	1	0	1	1	1



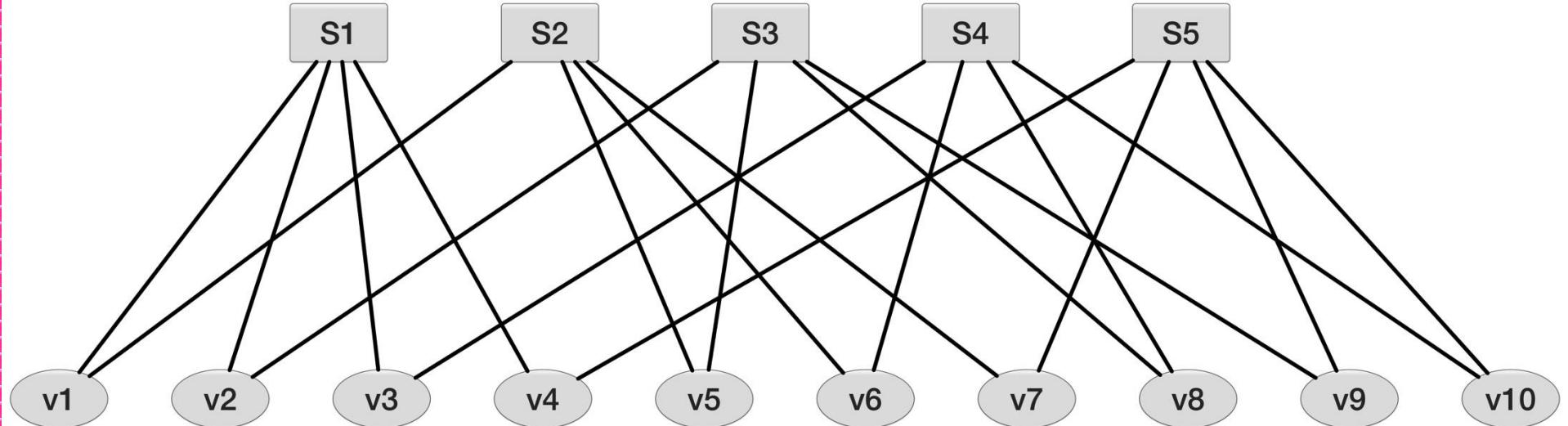
$H =$

1	1	1	1	0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	0	0	0	0	0
0	1	0	0	1	0	0	1	1	0	0	0
0	0	1	0	0	1	0	1	0	1	0	1
0	0	0	1	0	0	1	0	1	0	1	1



$H =$

1	1	1	1	0	0	0	0	0	0	0
1	0	0	0	1	1	1	0	0	0	0
0	1	0	0	1	0	0	1	1	0	0
0	0	1	0	0	1	0	1	0	1	1
0	0	0	1	0	0	1	0	1	1	1



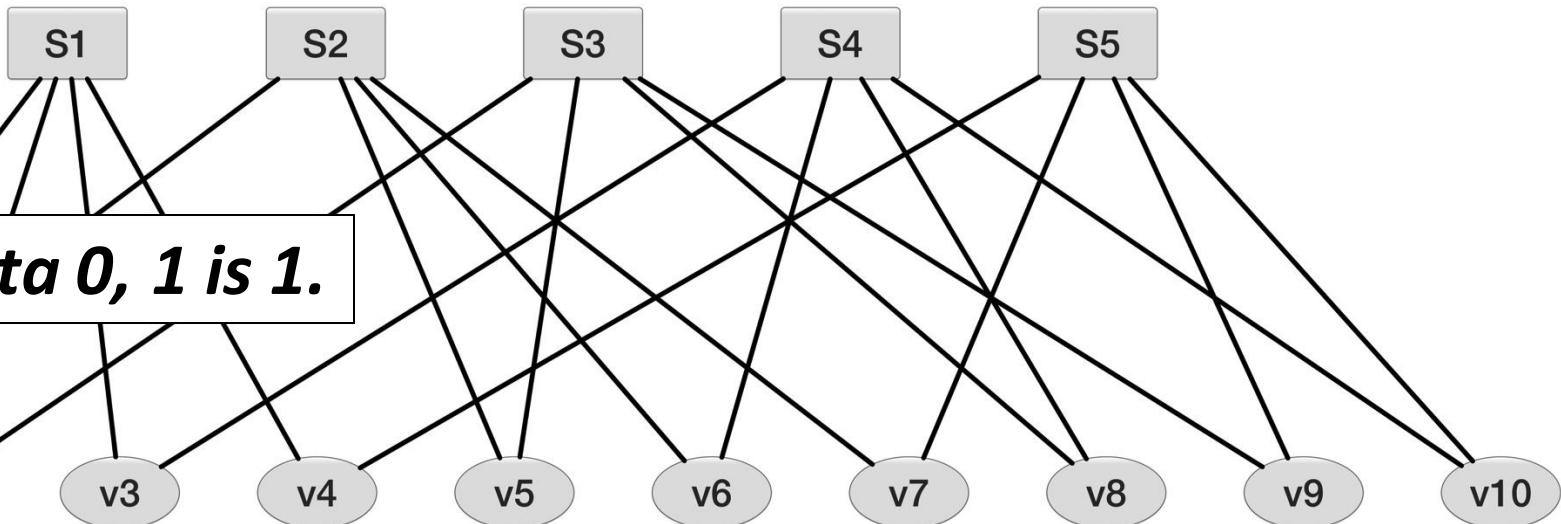


Are you Regular?

- A regular code means the number of 1s is constant for all rows, and constant for all columns. All code bits are involved in the same number of parity checks, each parity check has the same number of code bits coming in.
- But being irregular gives advantages!

$H =$

1	1	1	1	0	0	0	0	0	0	0
1	0	0	0	1	1	1	0	0	0	0
0	1	0	0	1	0	0	1	1	0	0
0	0	1	0	0	1	0	1	0	1	1
0	0	0	1	0	0	1	0	1	1	1



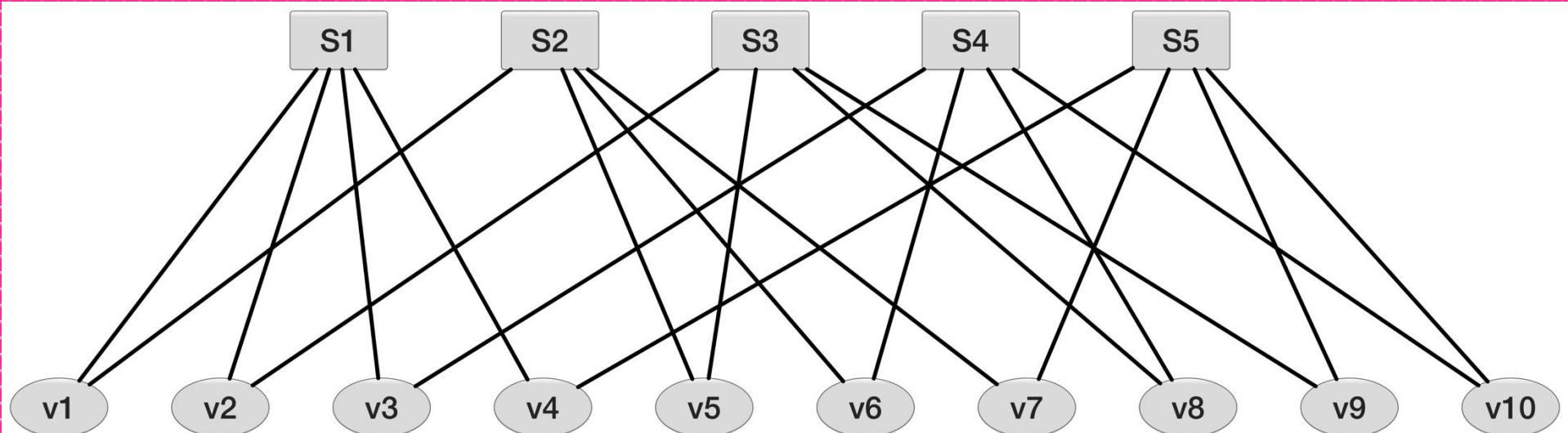
-1.1 -1.2 -0.9 0.3 -0.8 -1.5 -1.1 -1.2 -0.2 -1.1

$H =$

1 1 1 0 0 0 0 0 0 0
1 0 0 1 0 0 0 0 0 0
1 0 0 0 1 0 0 0 0 0
0 1 0 0 1 0 0 0 0 0
0 0 1 0 0 1 0 0 0 0
0 0 0 1 0 0 1 0 0 0
0 0 0 0 1 0 0 1 0 0
0 0 0 0 0 1 0 0 1 0
0 0 0 0 0 0 1 0 0 1
0 0 0 0 0 0 0 1 0 0

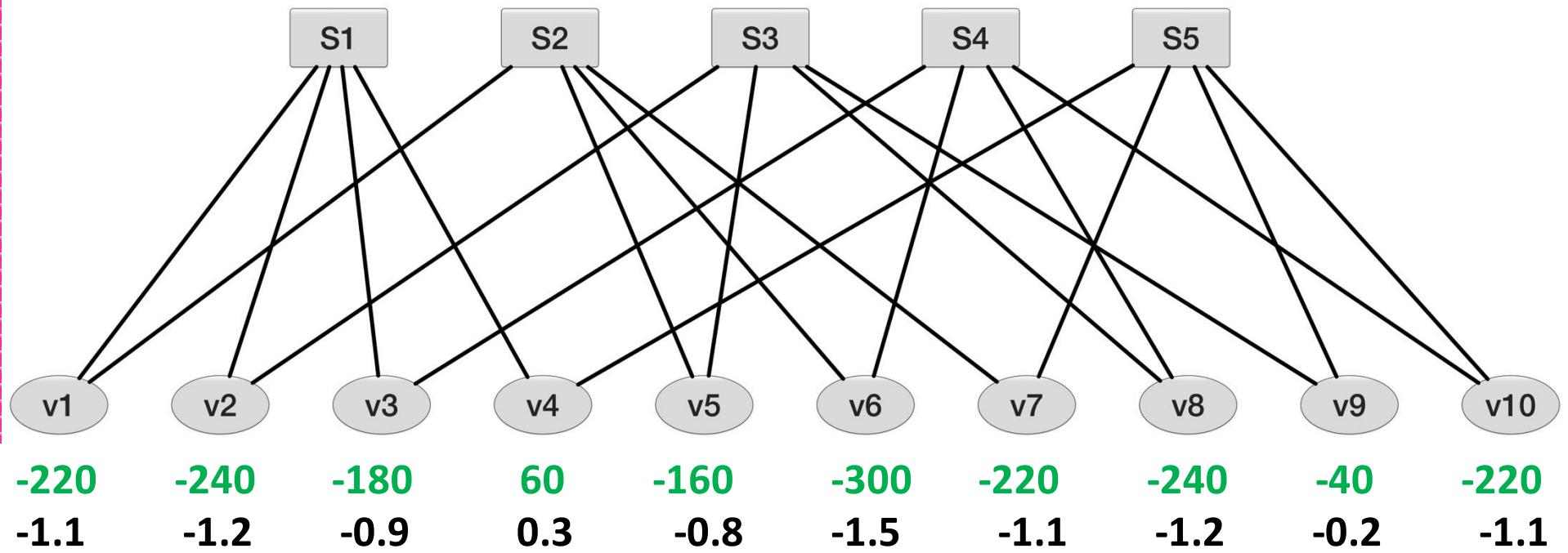
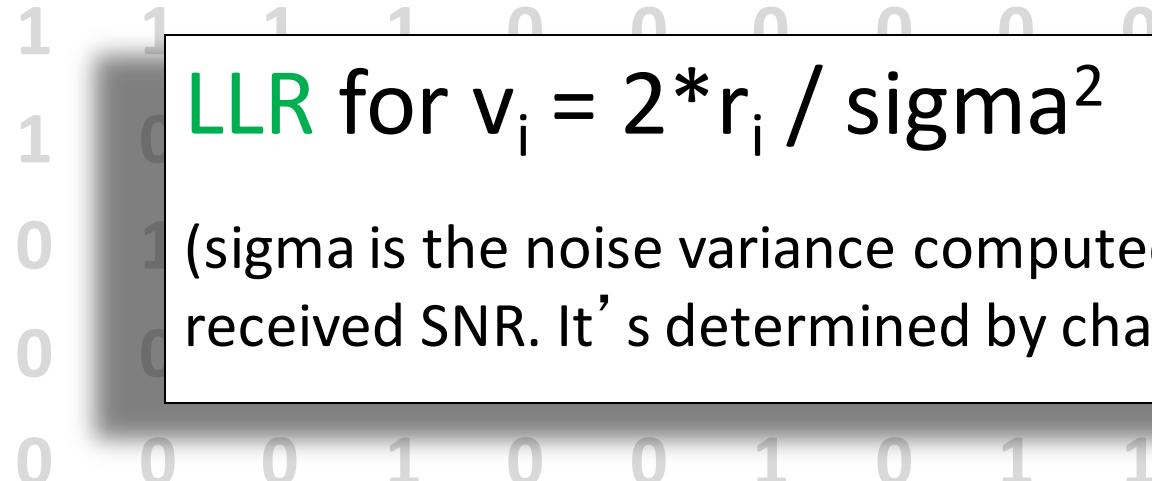
LLR for $v_i = 2 * r_i / \sigma^2$

(σ is the noise variance computed from the received SNR. It's determined by channel conditions.)
Here, let's set $\sigma = 0.1$



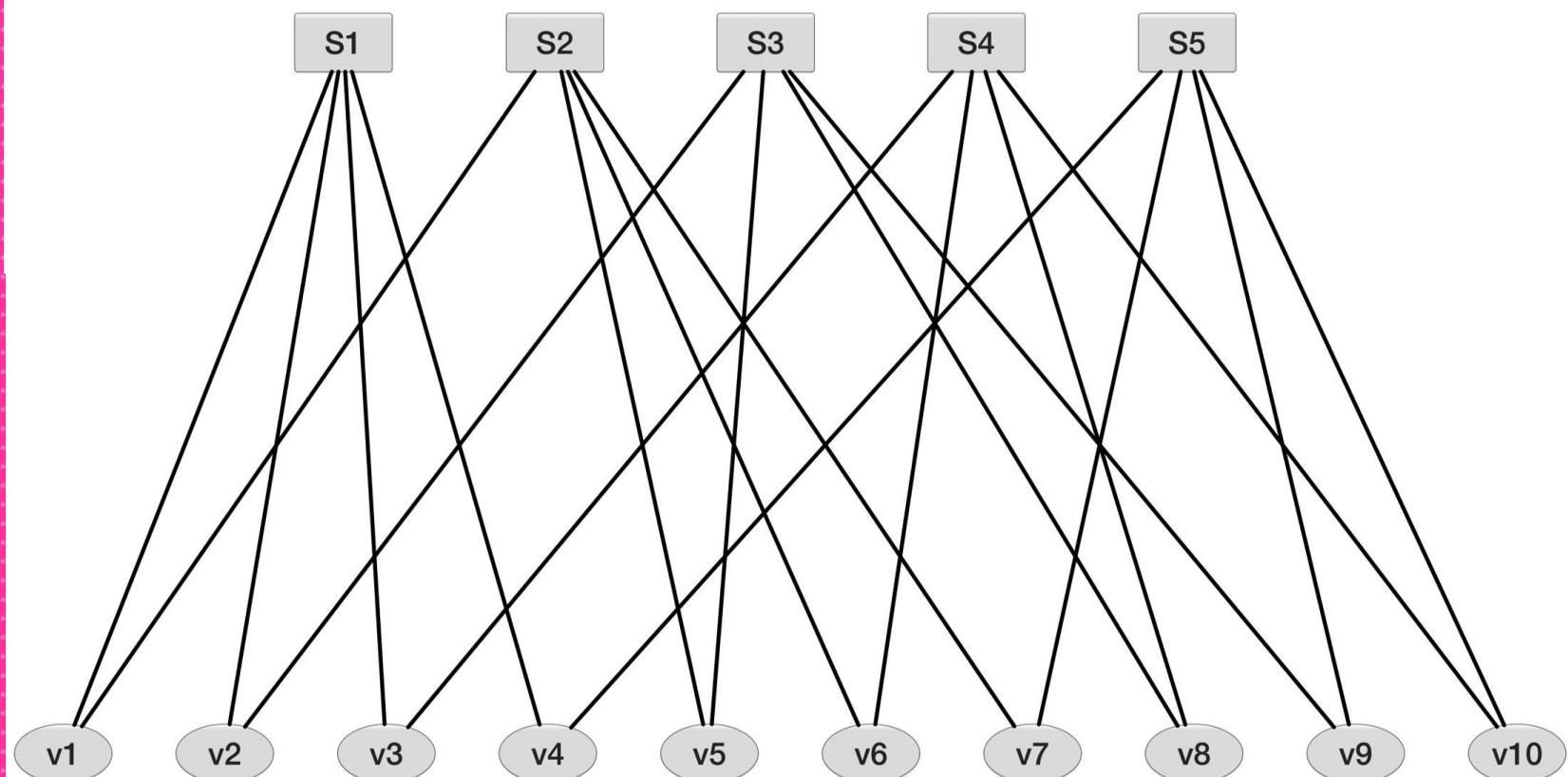
-1.1 -1.2 -0.9 0.3 -0.8 -1.5 -1.1 -1.2 -0.2 -1.1

H =



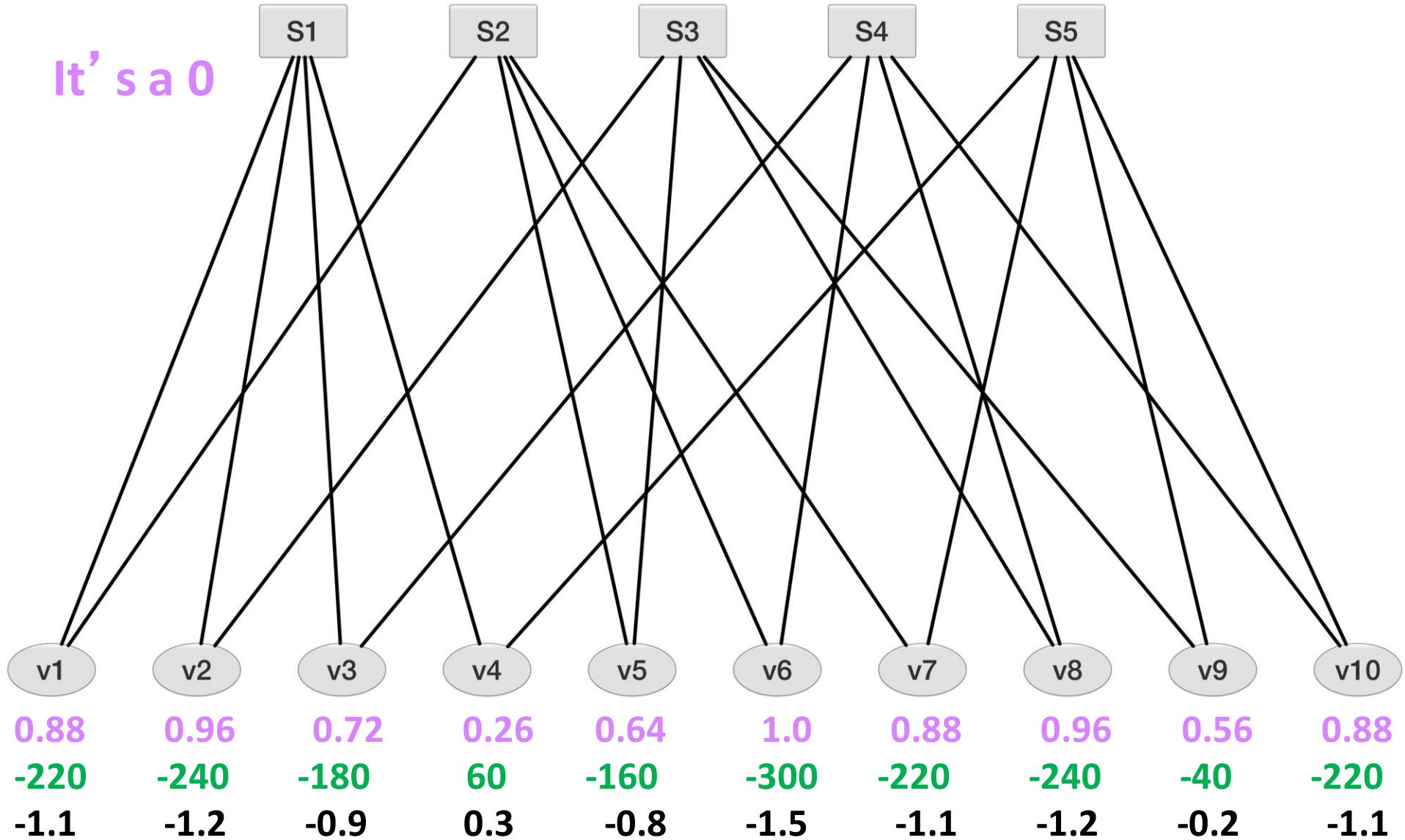
$H =$

1	1	1	1	0	0	0	0	0	0	0
1	0	0	0	1	1	1	0	0	0	0
0	1	0	0	1	0	0	1	1	0	

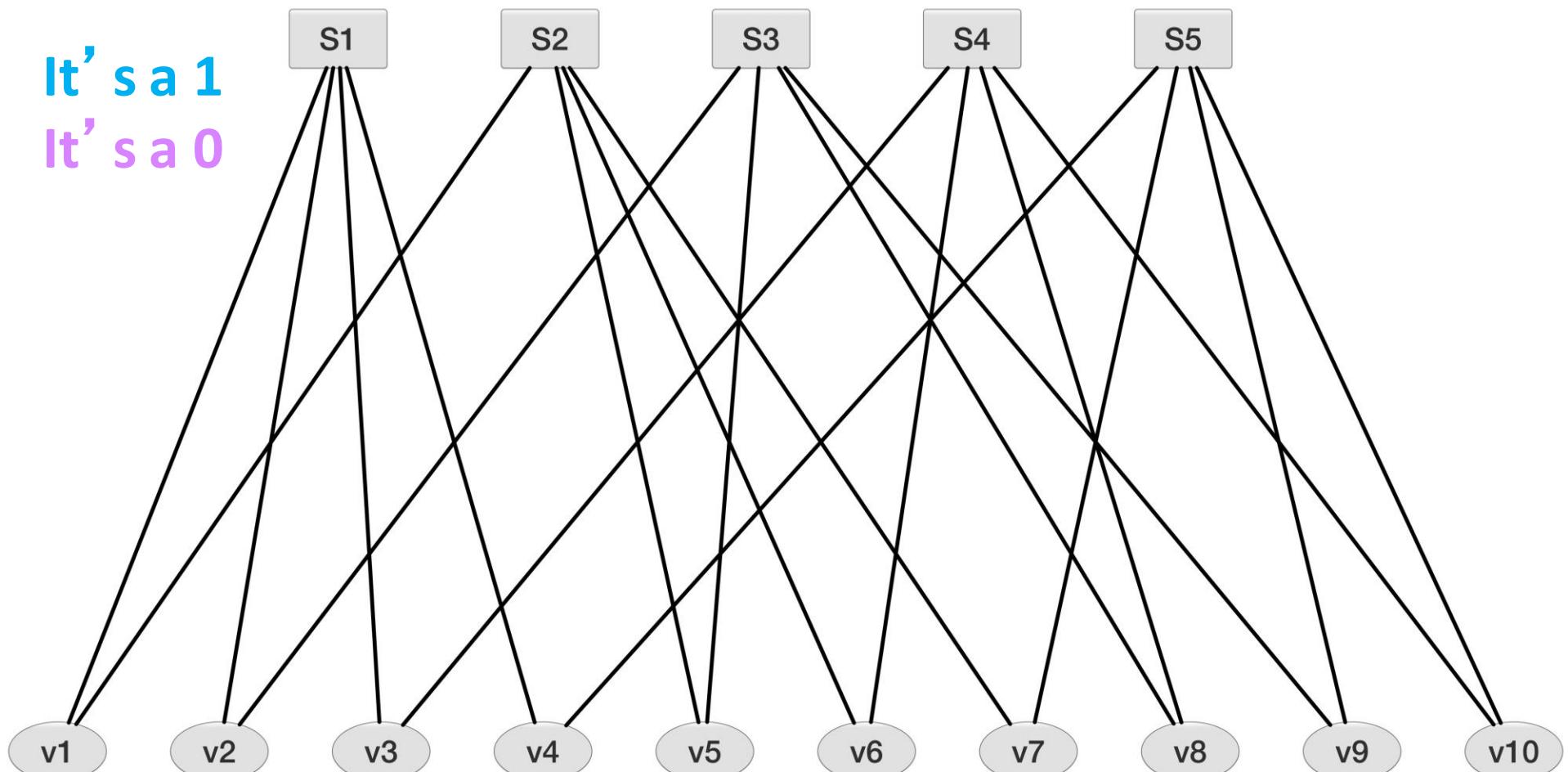


-220	-240	-180	60	-160	-300	-220	-240	-40	-220
-1.1	-1.2	-0.9	0.3	-0.8	-1.5	-1.1	-1.2	-0.2	-1.1

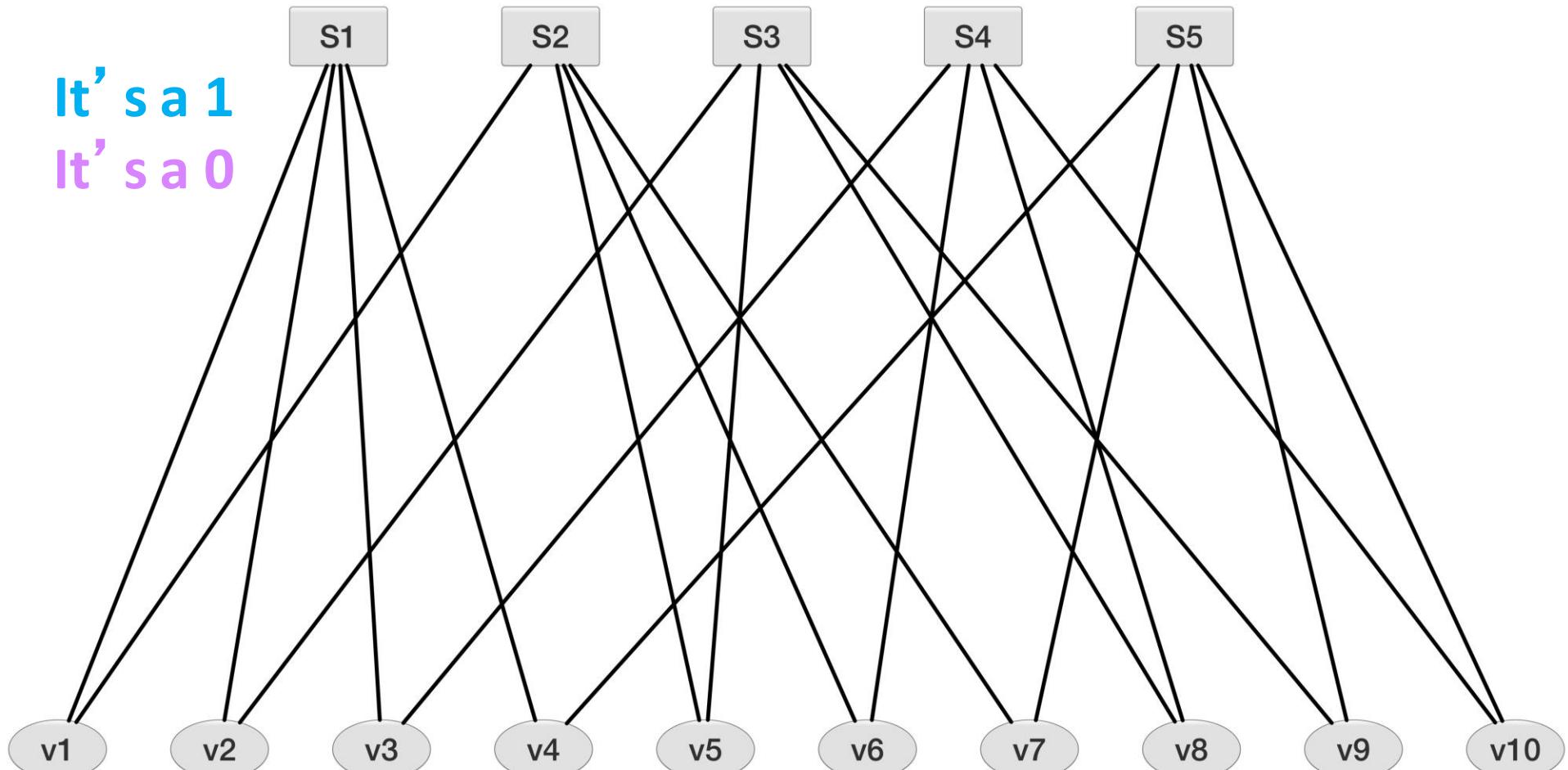
It's a 0



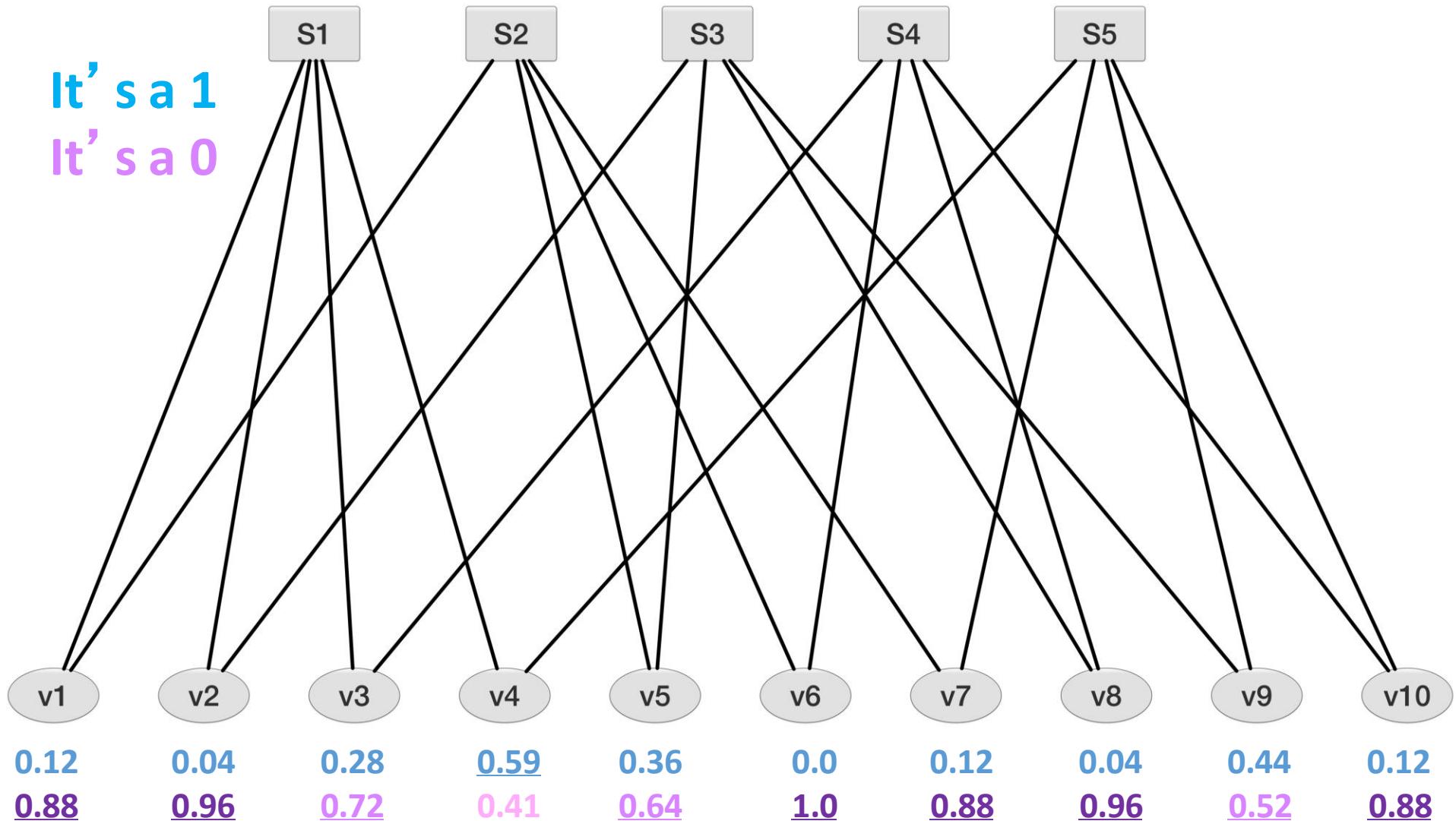
It's a 1
It's a 0



0.12	0.04	0.28	0.59	0.36	0.0	0.12	0.04	0.44	0.12
0.88	0.96	0.72	0.41	0.64	1.0	0.88	0.96	0.52	0.88
-220	-240	-180	60	-160	-300	-220	-240	-40	-220
-1.1	-1.2	-0.9	0.3	-0.8	-1.5	-1.1	-1.2	-0.2	-1.1

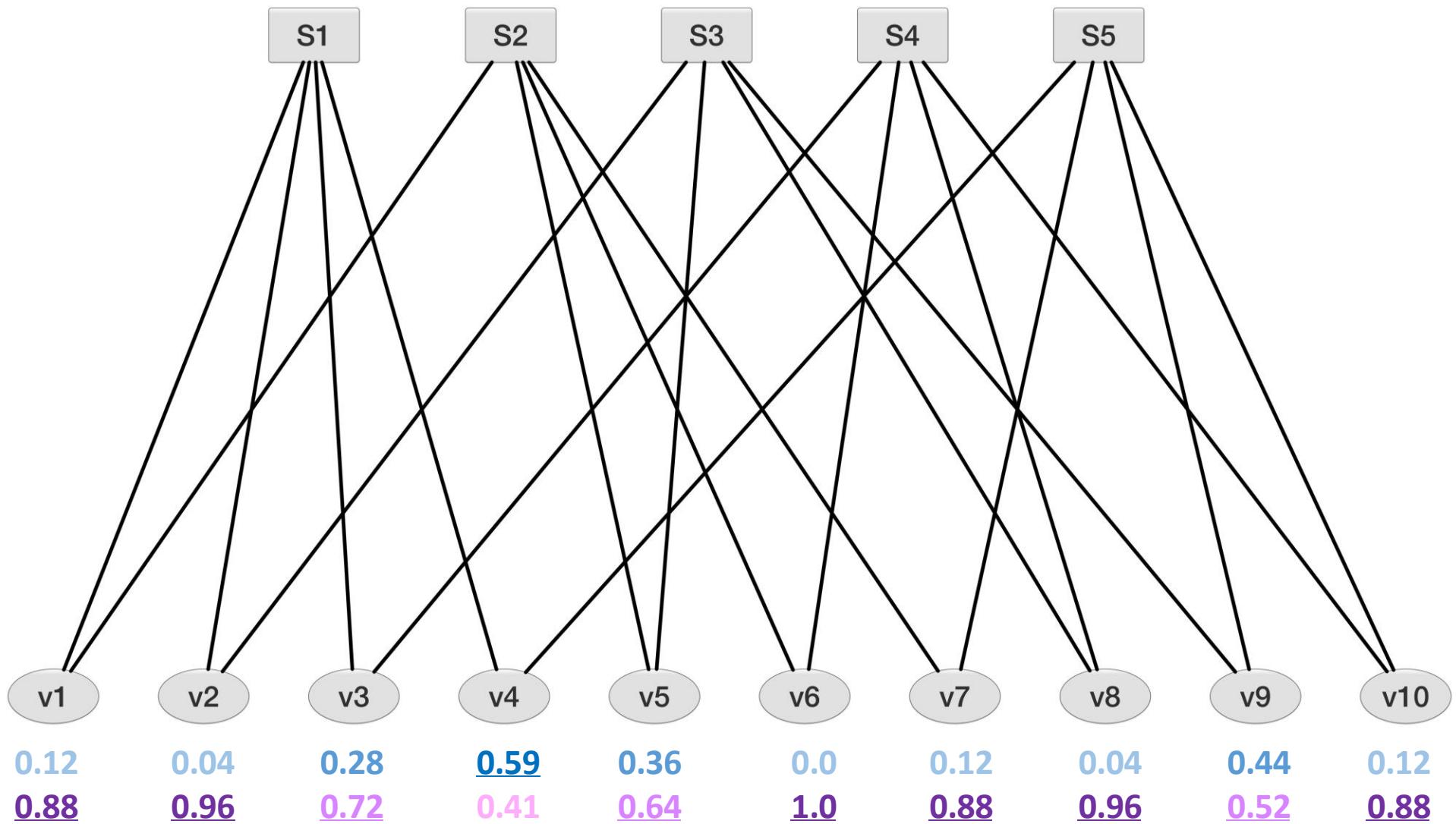


It's a 1
It's a 0

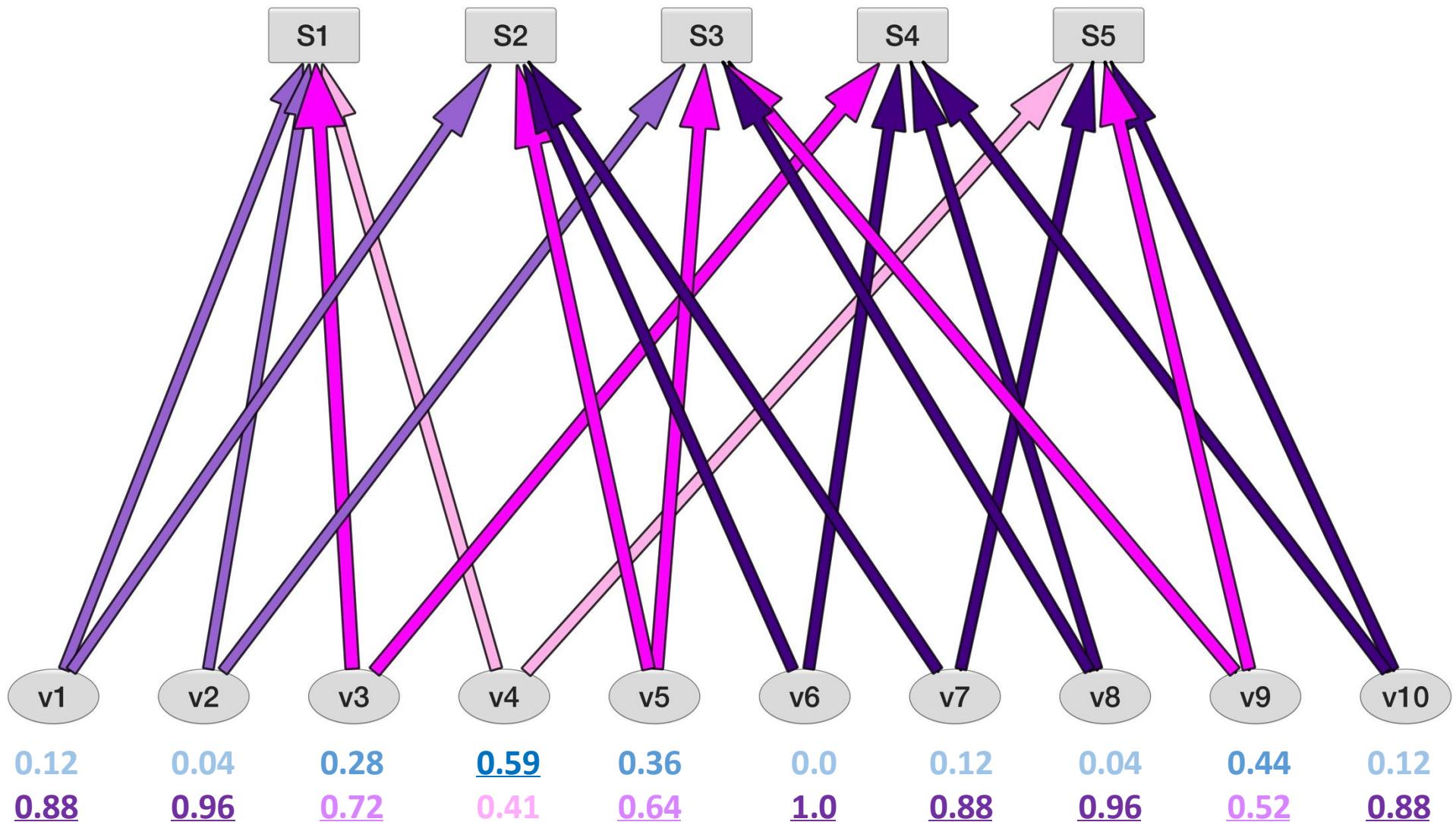


For each bit node we now have a probability it's a -1 and the probability that it is a 1. The two probabilities add to a total of 1. -1 is data 0, 1 is 1.

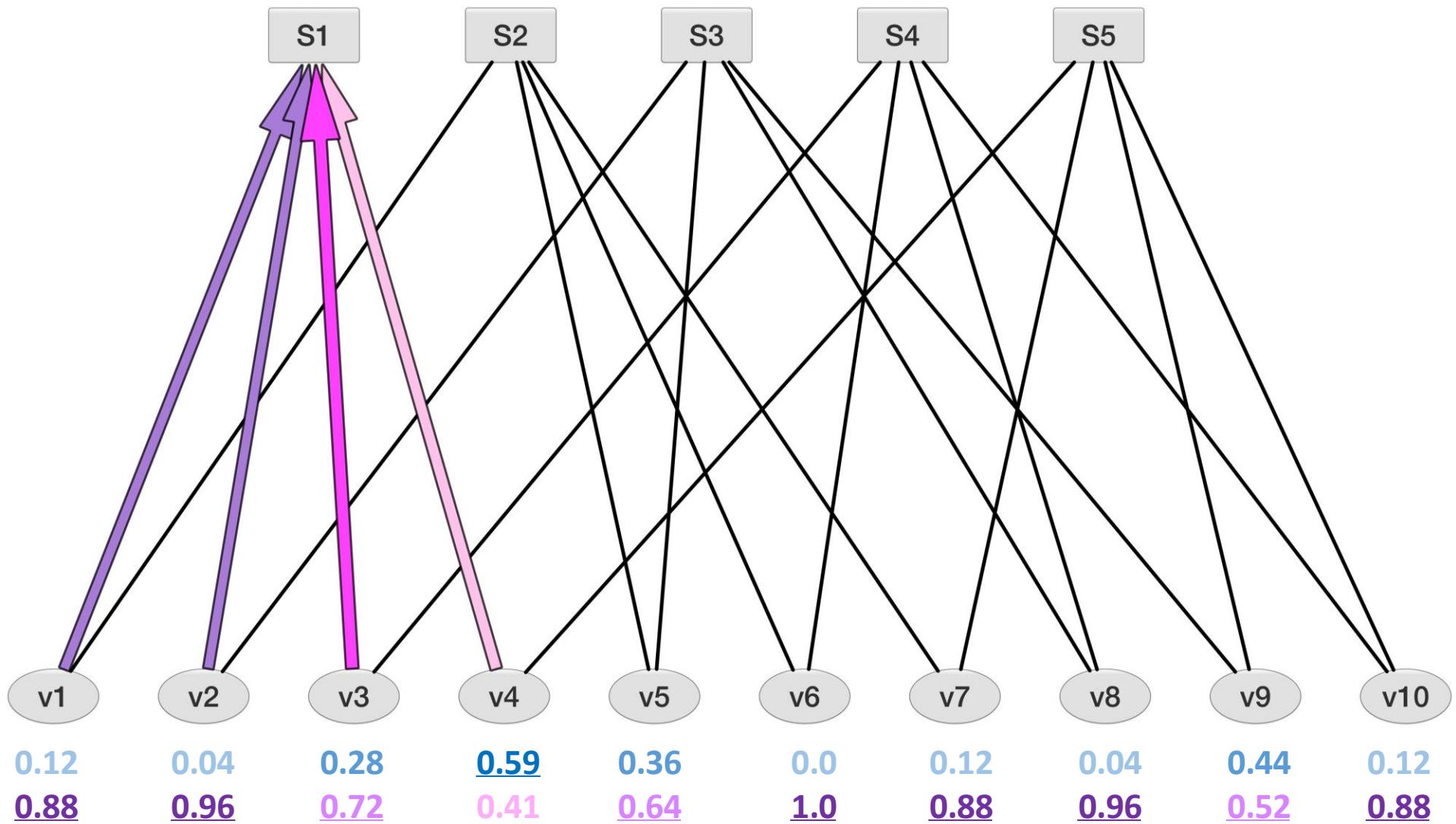
We send these probabilities to the check sum nodes!



We send these probabilities to the check sum nodes!

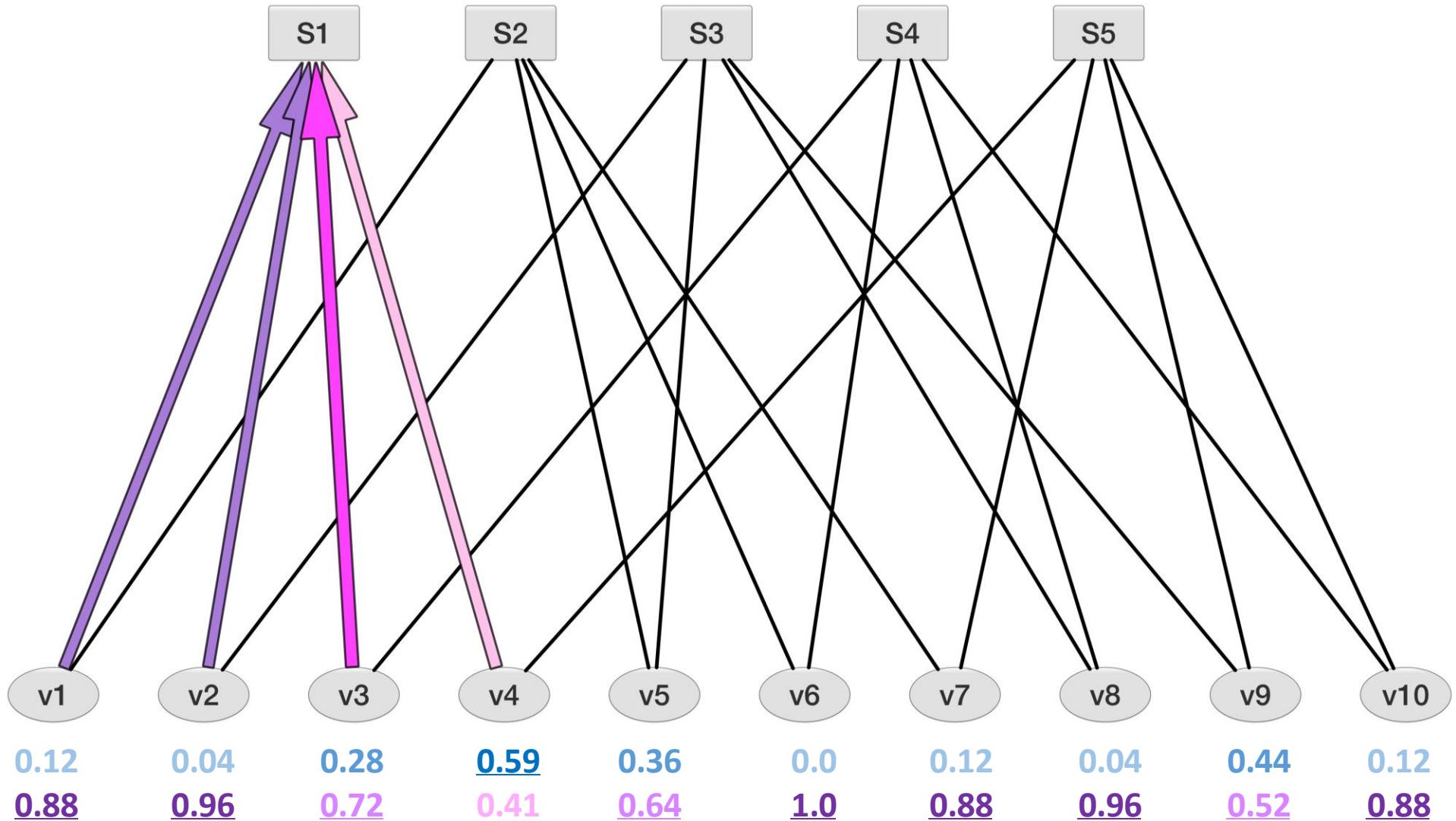


We send these probabilities to the check sum nodes!

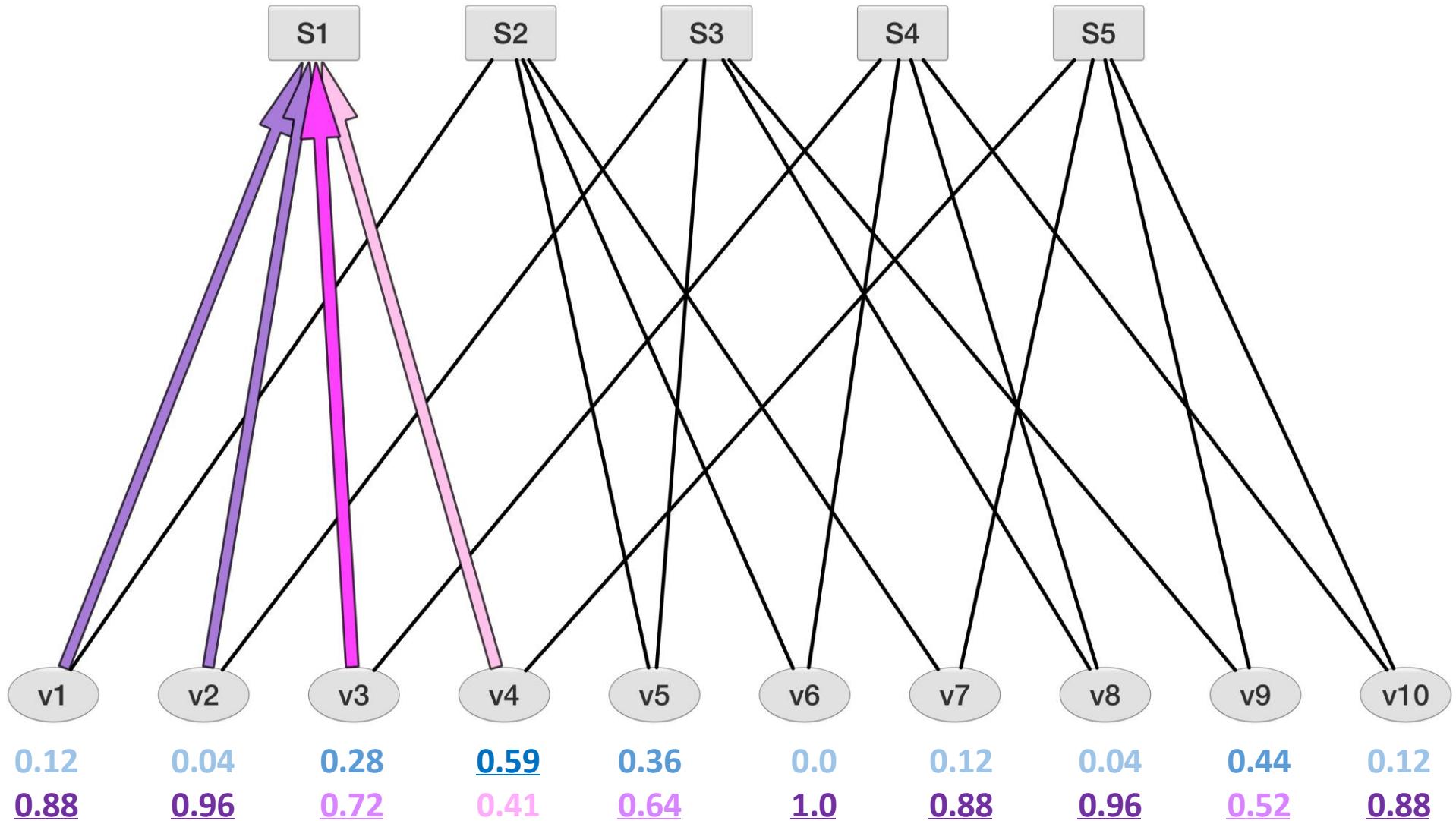


Now we do some work in the check sum nodes.

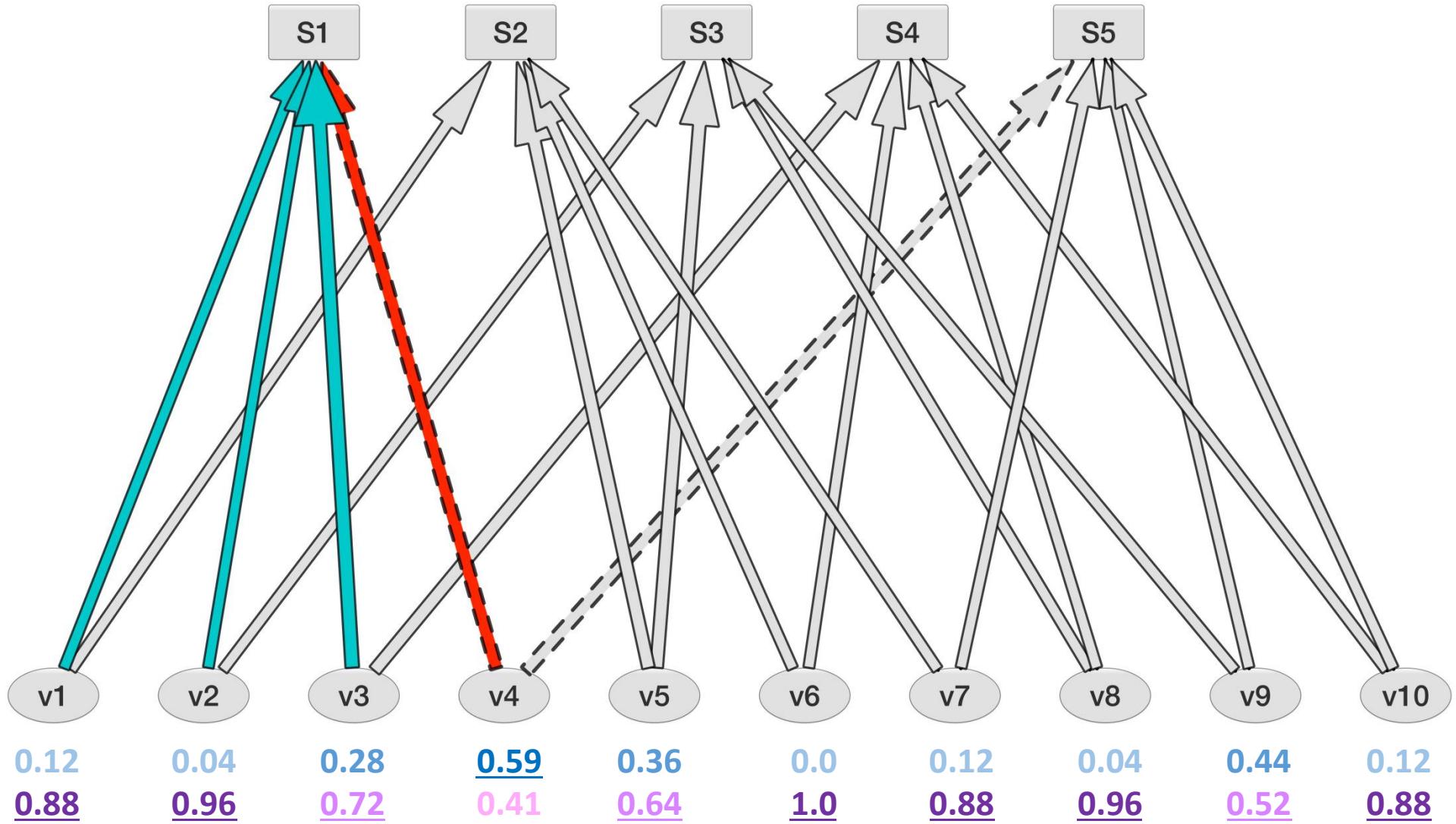
Each node is one row of the parity check matrix H.



Each S node should sum to zero, if the codeword is valid.
That takes an even number of 1's. (0 is an even number)

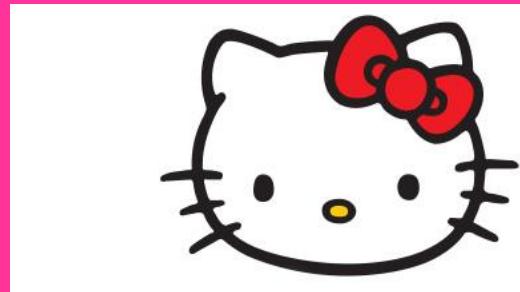
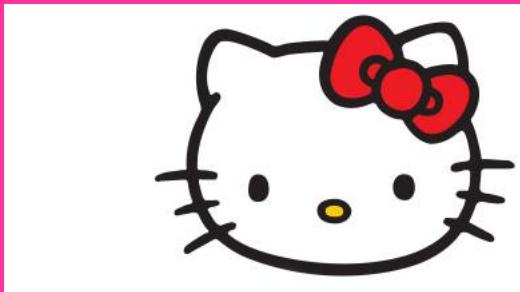


Let's look at S1 to see what we do for this side of the iteration.



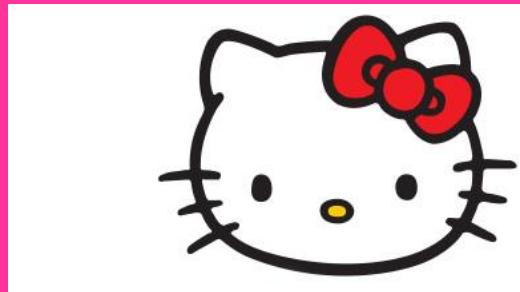


You know how it is.



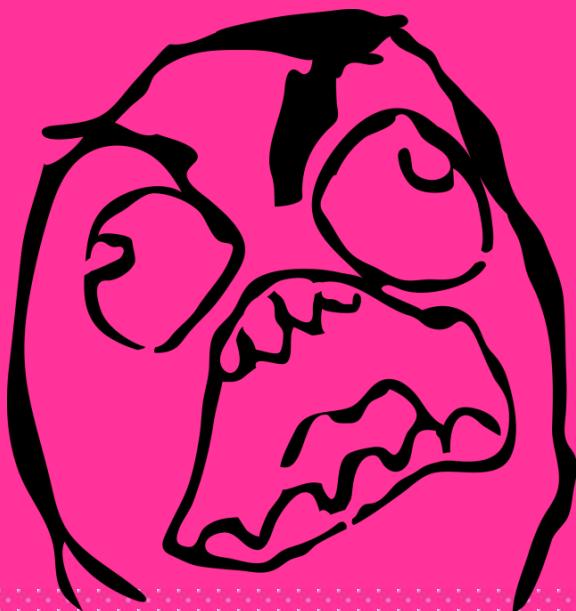
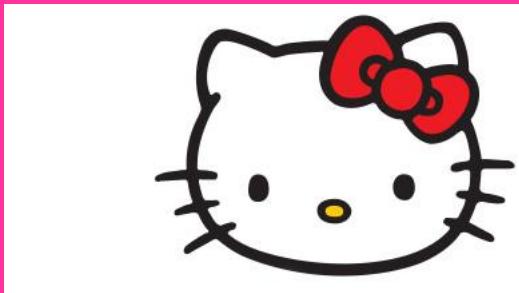
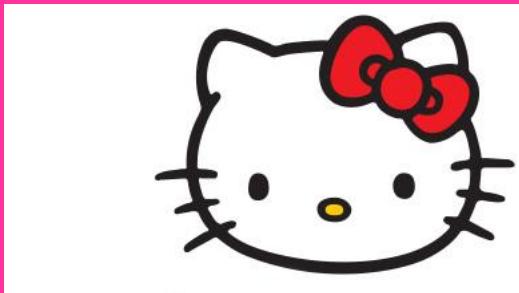


You have an even # of children.

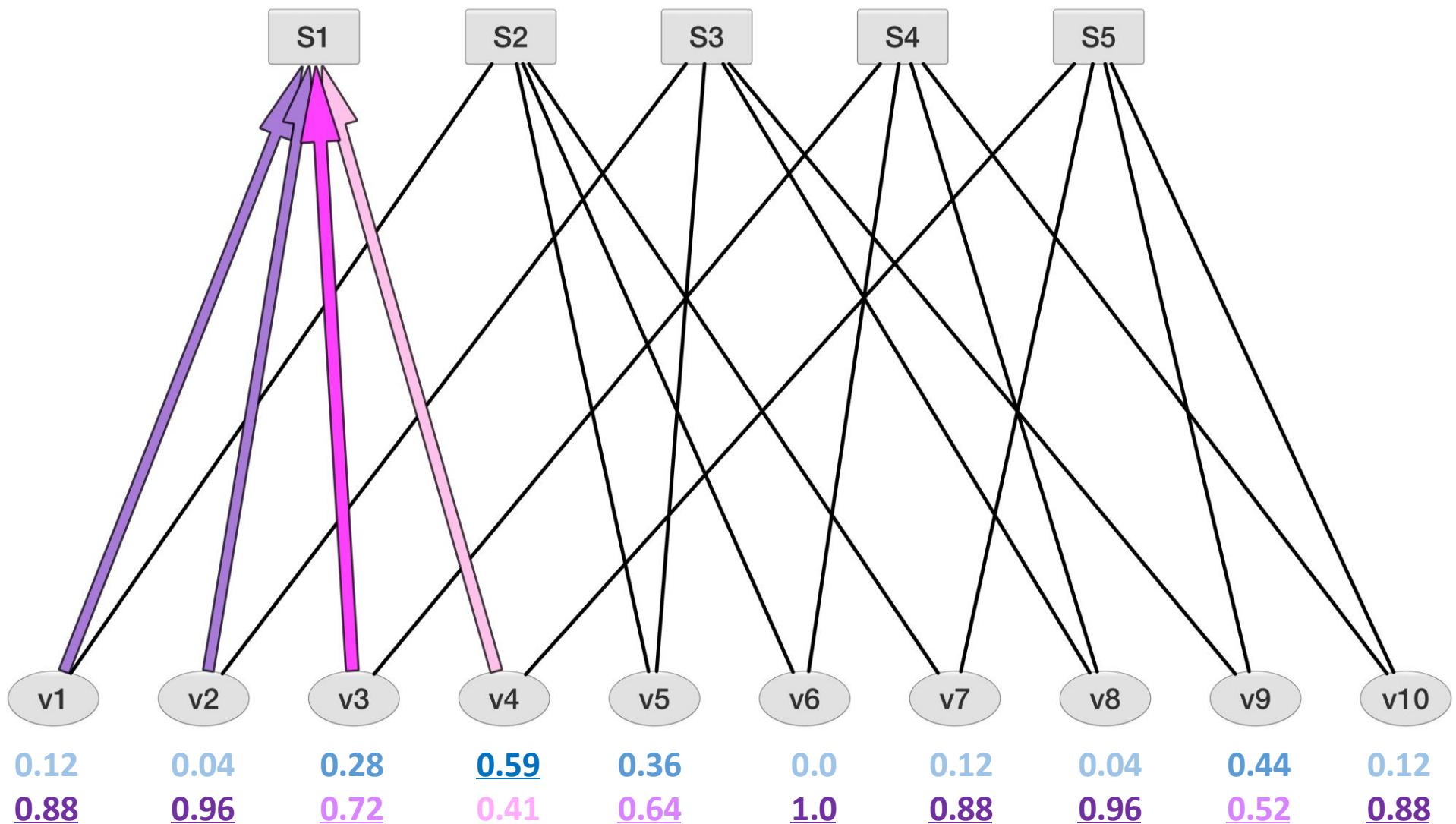




And one doesn't cooperate.



S1 tells each bit node how probable it is that they're a 1.
It's like a parent telling each child what the other children
think of them, and maybe how they should change.





How does this happen?

- Hey, **variable node 4**, let's assume you're a 1.
- If you're a 1, then there's either one more 1 besides you, or there are three other 1s. You're all 1s.
- You can't be the only 1.
- We can't have you be 1, and two others be 1. That would be three 1s.



Written out in Math

$$\begin{aligned} p(v_4=1) = & p(v_1)*(1-p(v_2))*(1-p(v_3)) + \\ & p(v_2)*(1-p(v_1))*(1-p(v_3)) + \\ & p(v_3)*(1-p(v_1))*(1-p(v_2)) + \\ & p(v_1)*p(v_2)*p(v_3) \end{aligned}$$



Written out in Math

$$\begin{aligned} p(v_4=1) = & p(v_1)*(1-p(v_2))*(1-p(v_3)) + \\ & p(v_2)*(1-p(v_1))*(1-p(v_3)) + \\ & p(v_3)*(1-p(v_1))*(1-p(v_2)) + \\ & p(v_1)*p(v_2)*p(v_3) \end{aligned}$$

Likelihood of v_i being equal to data 1



Written out in Math

$$\begin{aligned} p(v_4=1) = & p(v_1)*(1-p(v_2))*(1-p(v_3)) + \\ & p(v_2)*(1-p(v_1))*(1-p(v_3)) + \\ & p(v_3)*(1-p(v_1))*(1-p(v_2)) + \\ & p(v_1)*p(v_2)*p(v_3) \end{aligned}$$

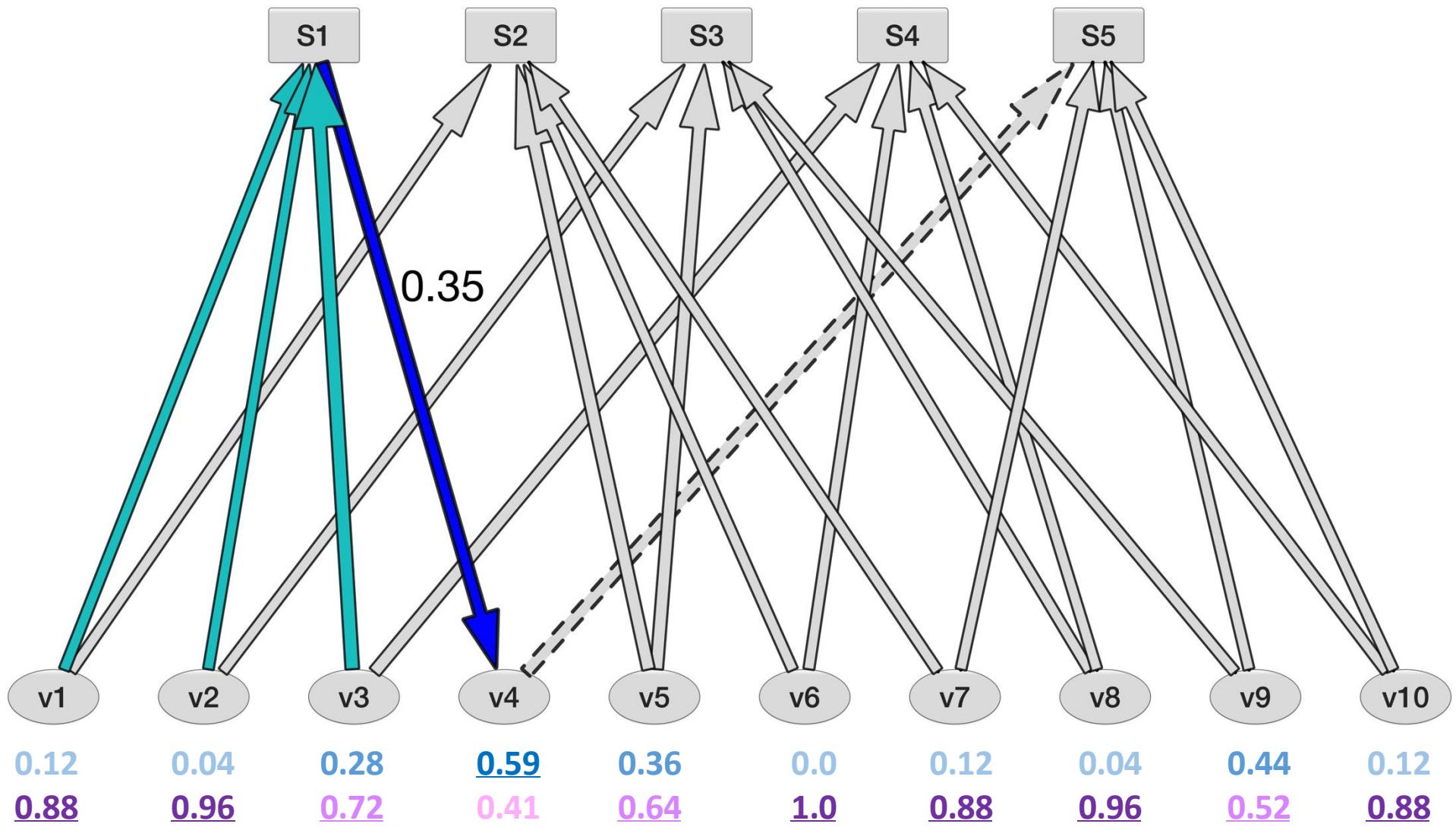
Each bit node gets a message formed from the probabilities from the other snitches. Nodes! Other bit nodes!



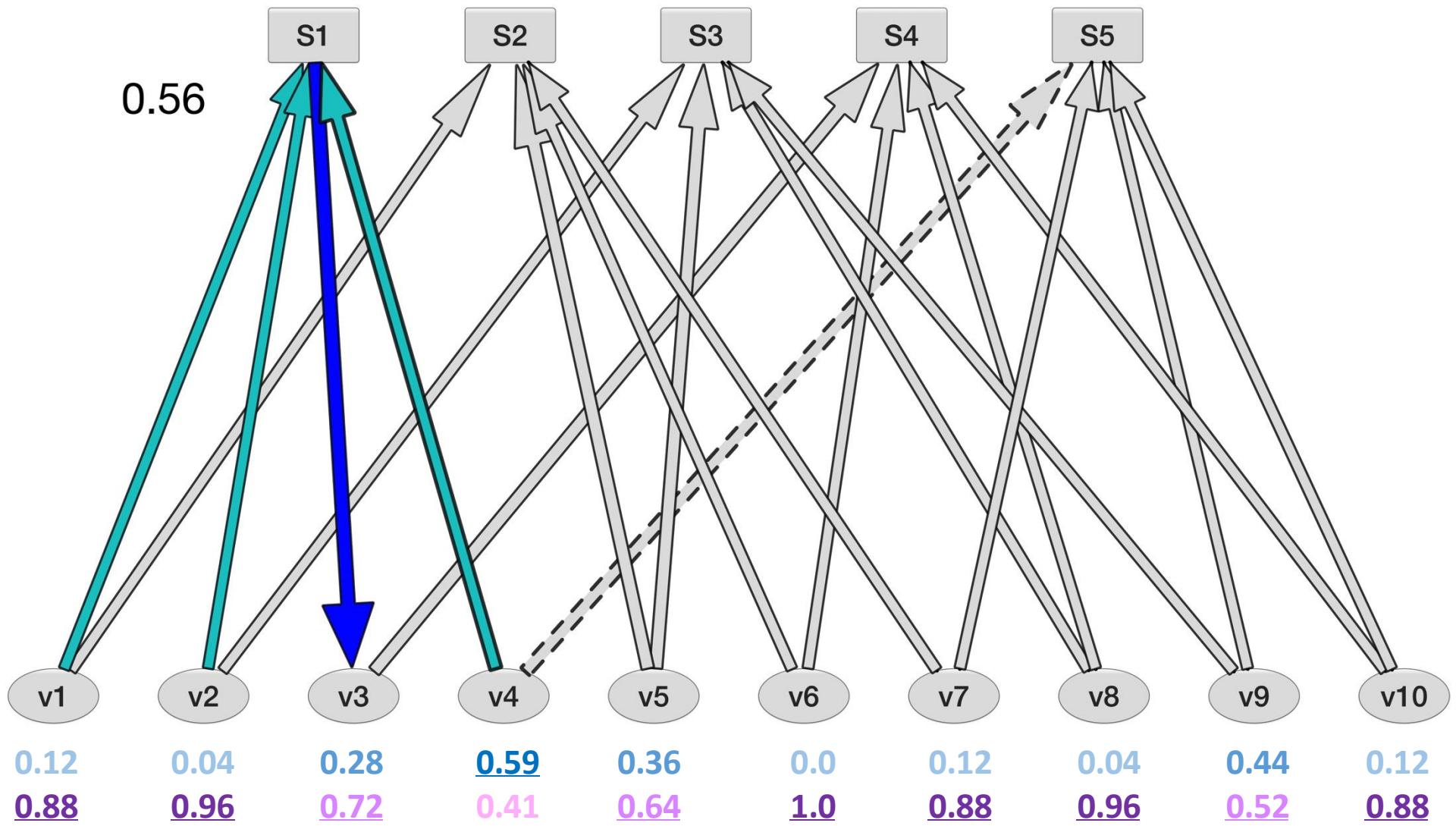
You have been weighed.

- So our result back to you, node 4, is 0.35
- You better shape up
- Ooo ooo ooo
- Because you need a plan

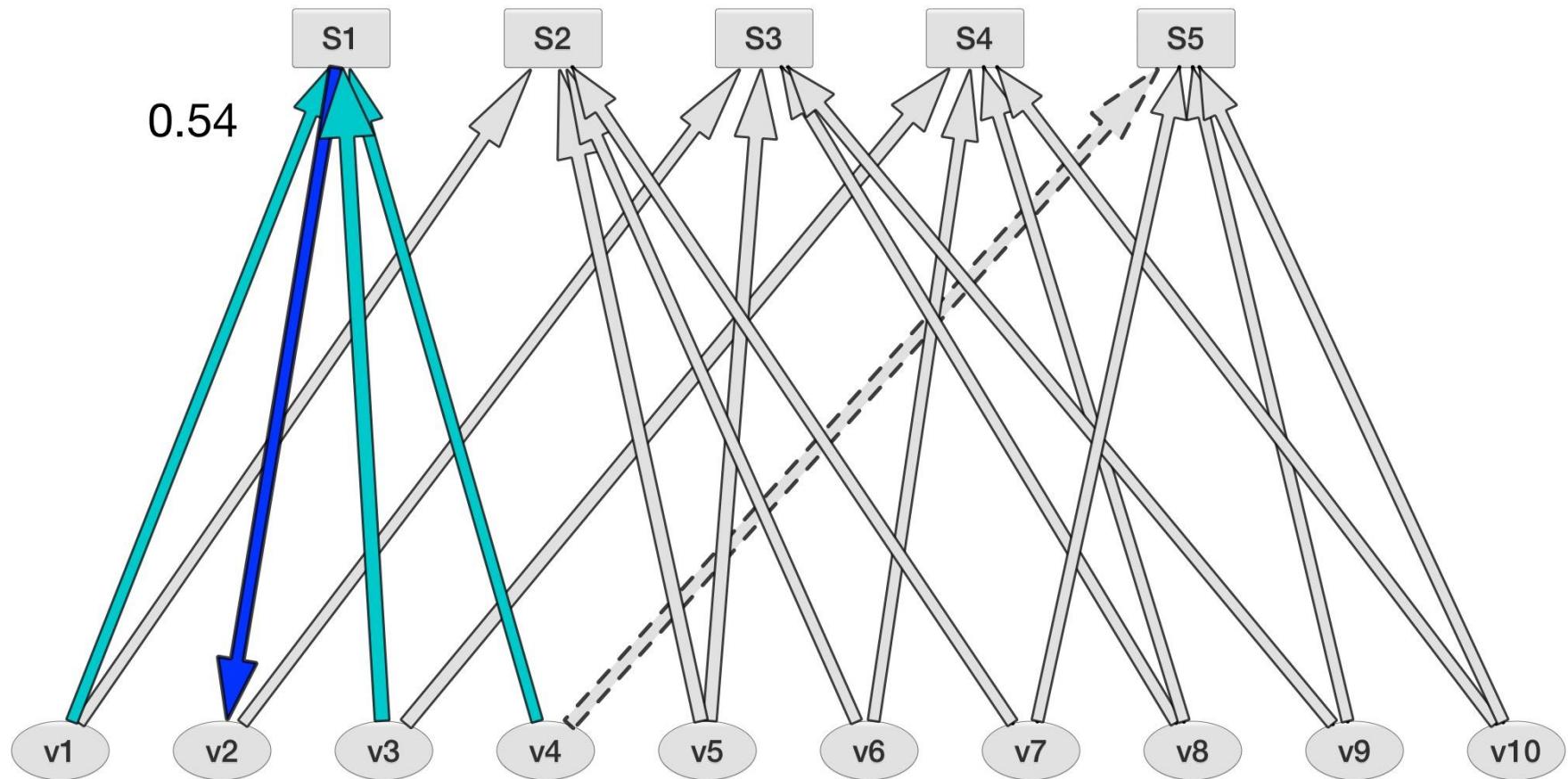
v4 updated from v1, v2, v3



v3 updated from v1, v2, v4

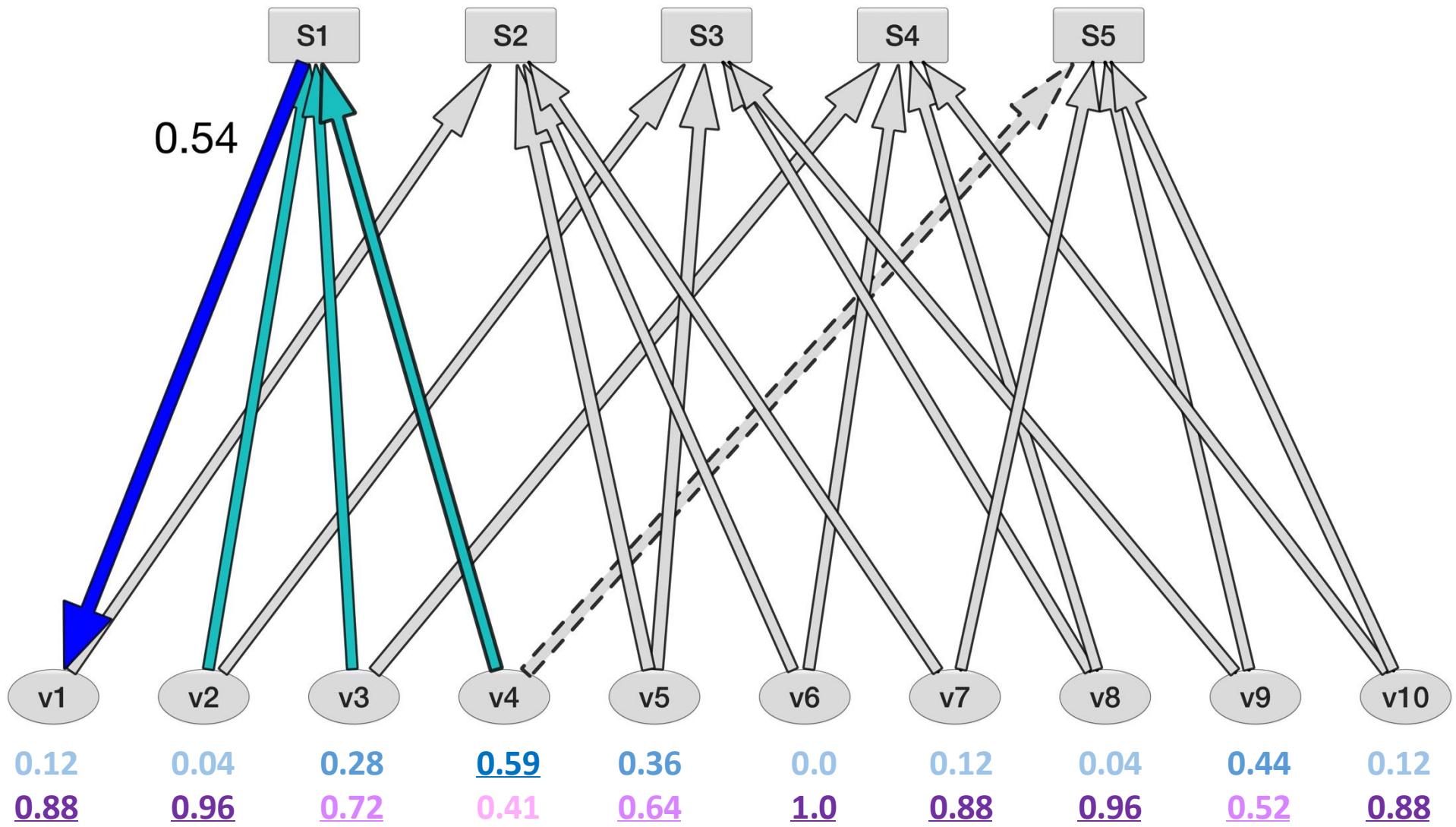


v2 updated from v1, v4, v3

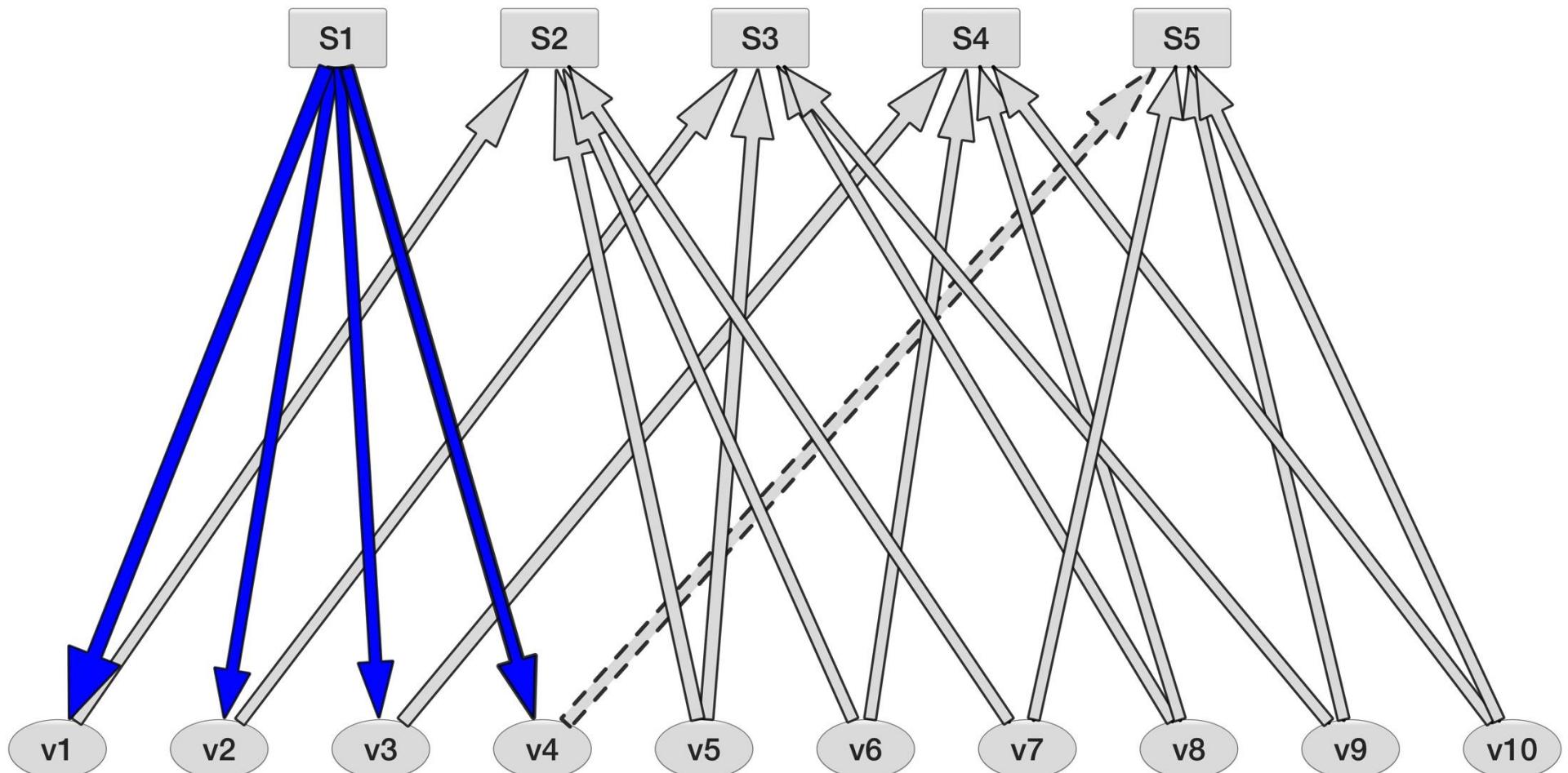


0.12	0.04	0.28	0.59	0.36	0.0	0.12	0.04	0.44	0.12
0.88	0.96	0.72	0.41	0.64	1.0	0.88	0.96	0.52	0.88

v1 updated from v4, v2, v3

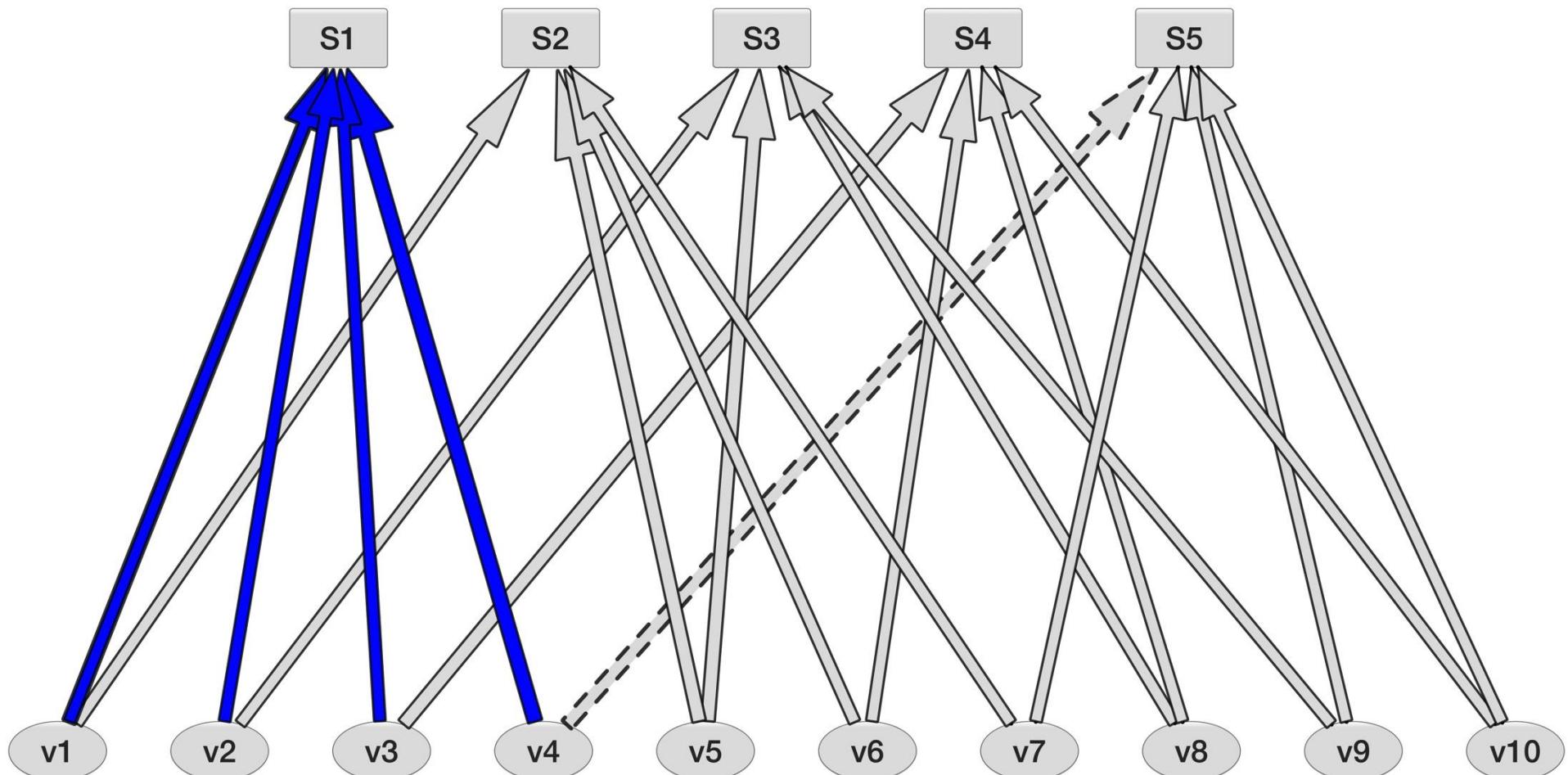


Bit nodes likelihoods updated.



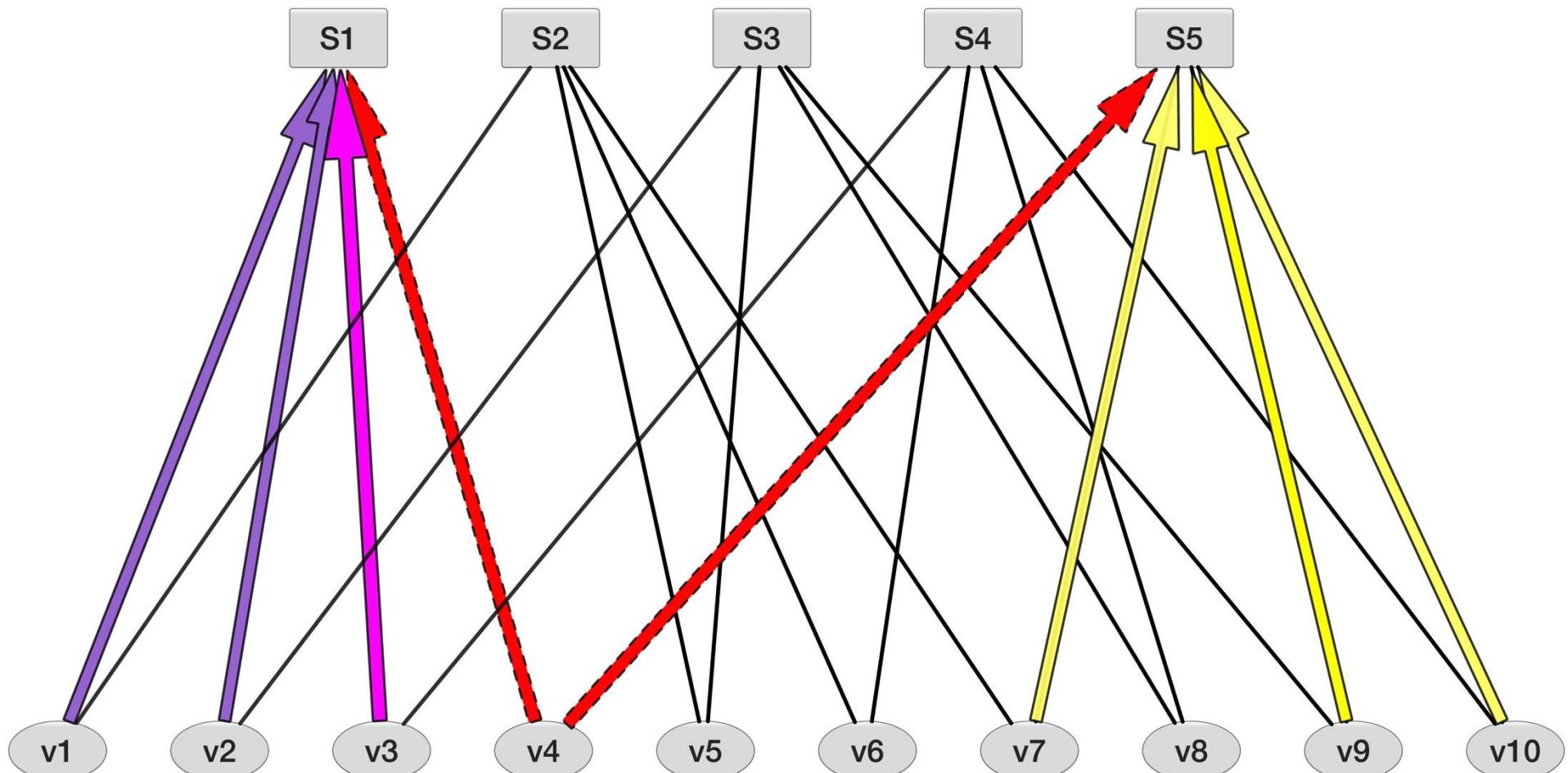
0.54	0.54	0.56	0.34							
0.12	0.04	0.28	0.59	0.36	0.0	0.12	0.04	0.44	0.12	
0.88	0.96	0.72	0.41	0.64	1.0	0.88	0.96	0.52	0.88	

The Real power is the cross connections.



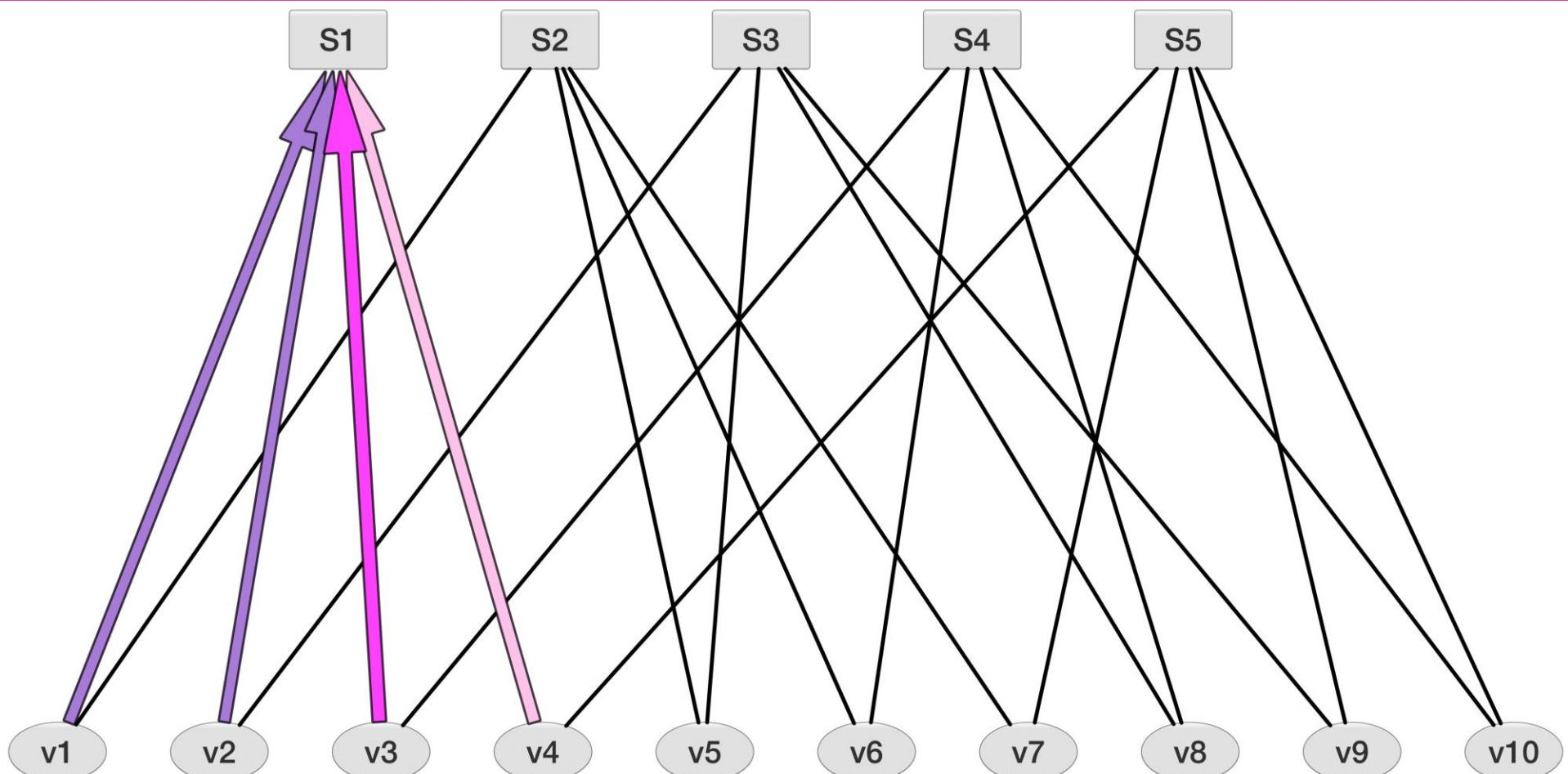
0.54	0.54	0.56	0.34							
0.12	0.04	0.28	0.59	0.36	0.0	0.12	0.04	0.44	0.12	
0.88	0.96	0.72	0.41	0.64	1.0	0.88	0.96	0.52	0.88	

The Real power is the cross connections.



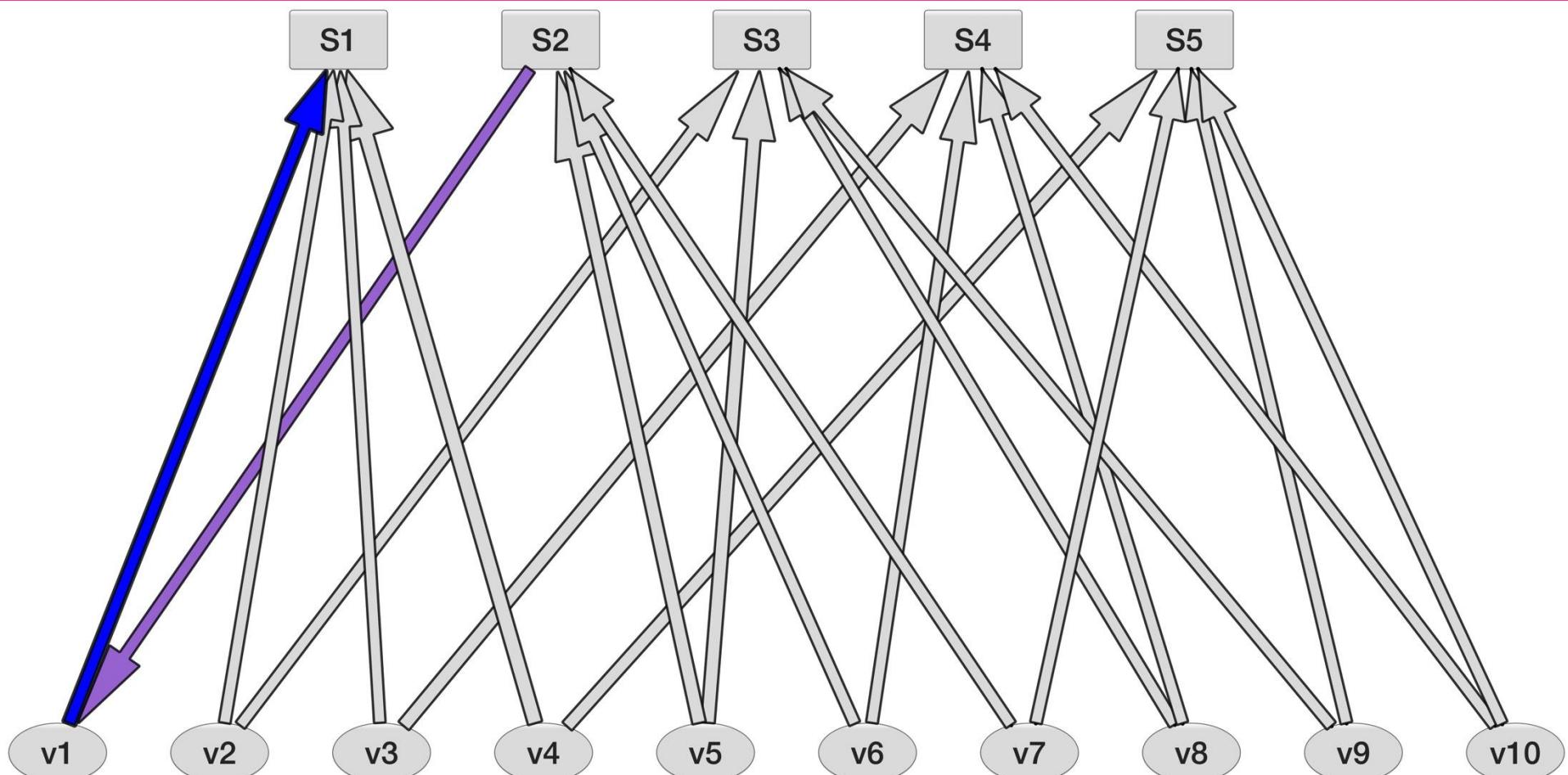
0.54	0.54	0.56	0.34							
0.12	0.04	0.28	<u>0.59</u>	0.36	0.0	0.12	0.04	0.44	0.12	
<u>0.88</u>	<u>0.96</u>	<u>0.72</u>	<u>0.41</u>	<u>0.64</u>	<u>1.0</u>	<u>0.88</u>	<u>0.96</u>	<u>0.52</u>	<u>0.88</u>	

Next, we send updated probabilities back.



0.54	0.54	0.56	0.34							
0.12	0.04	0.28	0.59	0.36	0.0	0.12	0.04	0.44	0.12	
0.88	0.96	0.72	0.41	0.64	1.0	0.88	0.96	0.52	0.88	

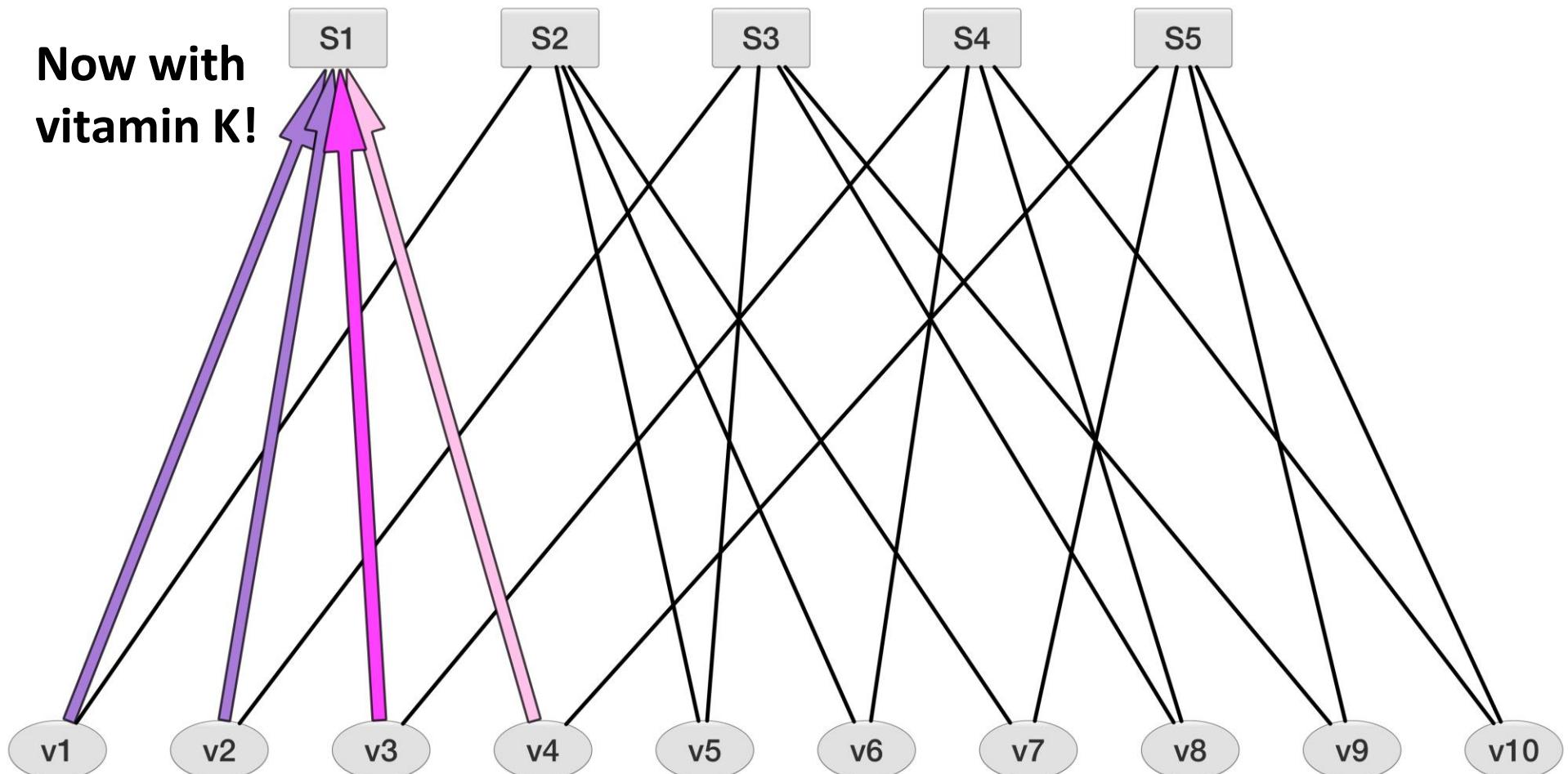
Hey look, info from other check nodes...



0.67	0.66	0.59	0.28							
0.12	0.04	0.28	0.59	0.36	0.0	0.12	0.04	0.44	0.12	
0.88	0.96	0.72	0.41	0.64	1.0	0.88	0.96	0.52	0.88	

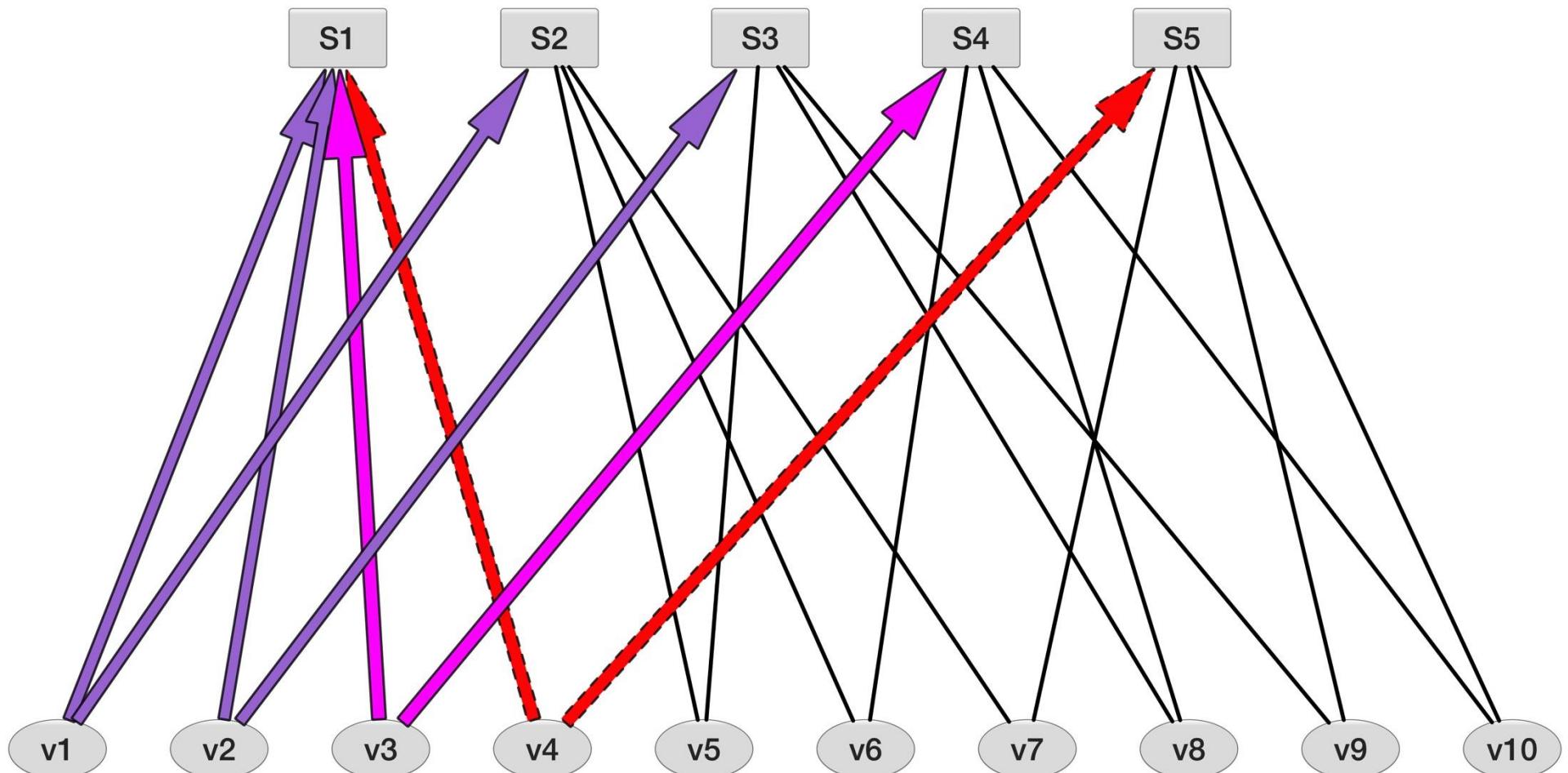
$$V1 = K * \text{LLR channel result} * p(a) * p(b)$$

Now with
vitamin K!



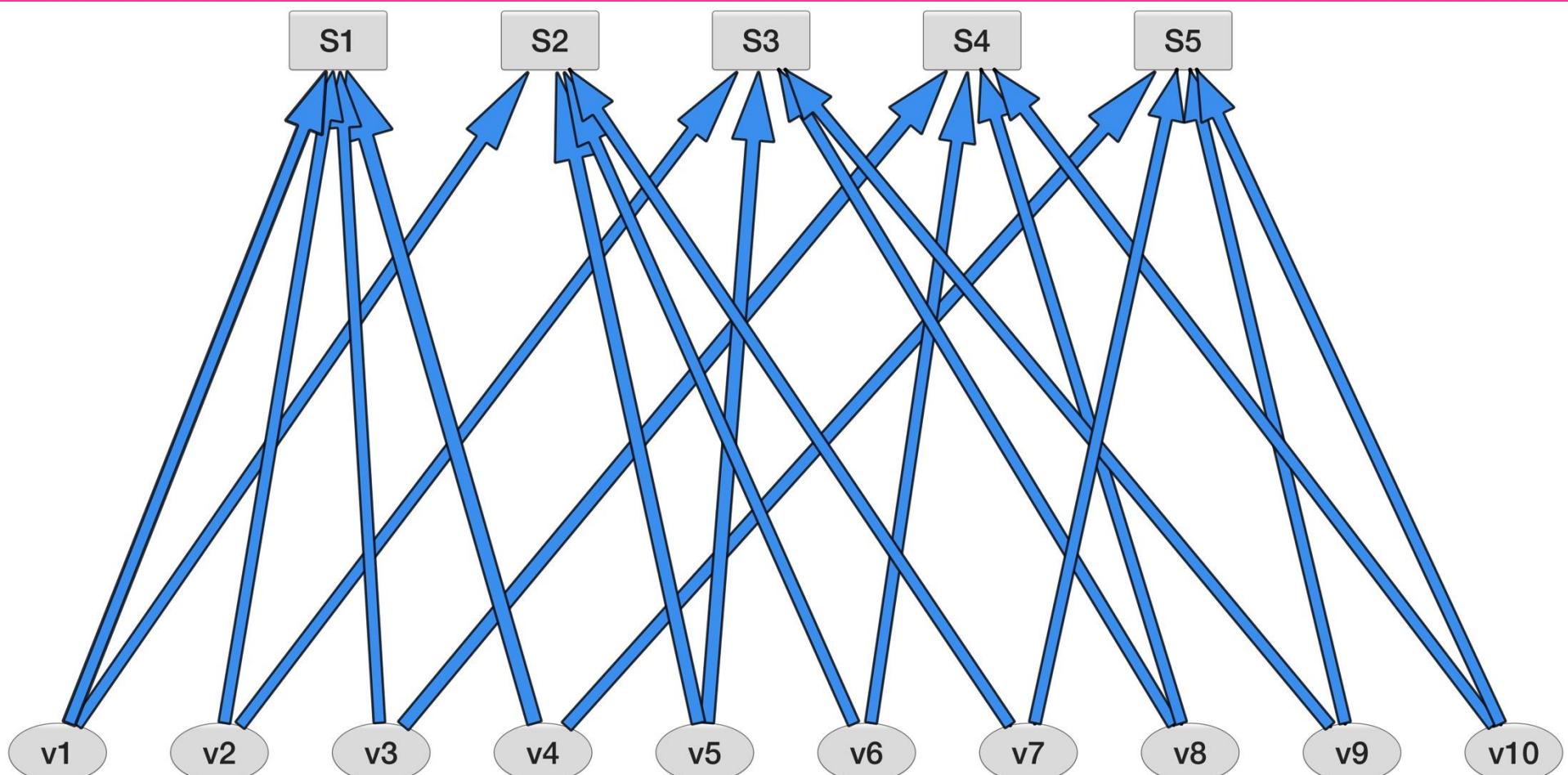
0.67	0.66	0.59	0.28						
0.12	0.04	0.28	0.59	0.36	0.0	0.12	0.04	0.44	0.12
0.88	0.96	0.72	0.41	0.64	1.0	0.88	0.96	0.52	0.88

All the parity check nodes affected by v1-v4.



0.67	0.66	0.59	0.28							
0.12	0.04	0.28	0.59	0.36	0.0	0.12	0.04	0.44	0.12	
0.88	0.96	0.72	0.41	0.64	1.0	0.88	0.96	0.52	0.88	

Everyone has snitched.



0.67	0.66	0.59	0.28	0.56	0.92	0.67	0.66	0.53	0.67
0.12	0.04	0.28	0.59	0.36	0.0	0.12	0.04	0.44	0.12
0.88	0.96	0.72	0.41	0.64	1.0	0.88	0.96	0.52	0.88



The beatings will continue

- Check node → variable node starts the iteration again.
- This back and forth continues until either a threshold probability is reached, or a maximum number of iterations has been reached.
- Convergence is an active area of study.



Sierra A

multiday hackfest and workshop
multiday hackfest and workshop

DVB-S2/X Block Party GRCon2018

Build and Test
DVB-S2 and DVB-S2X
Receivers in GNU Radio



ready to start now?

contact: @abraxas3d
w5nyv@yahoo.com

Thank You.

Terimakasih

