Computing Aggregated Quantities Efficiently in Large-Scale Dense Sensor Networks

Björn Andersson and Eduardo Tovar
Institute Polytechnic in Porto
Portugal
{bandersson,emt}@dei.isep.ipp.pt

Sensor Readings

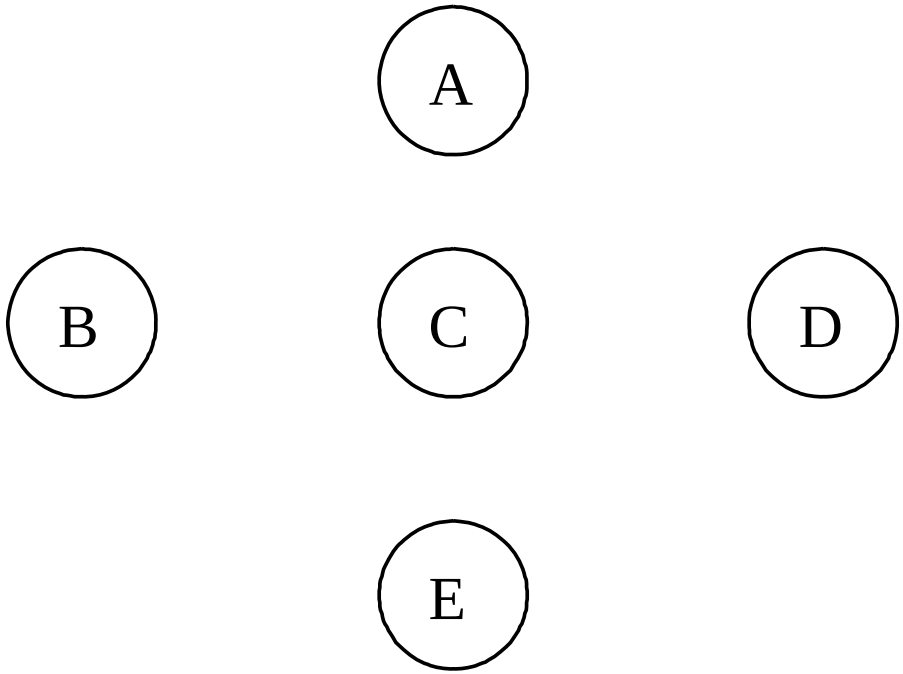
+

A prioritized MAC protocol

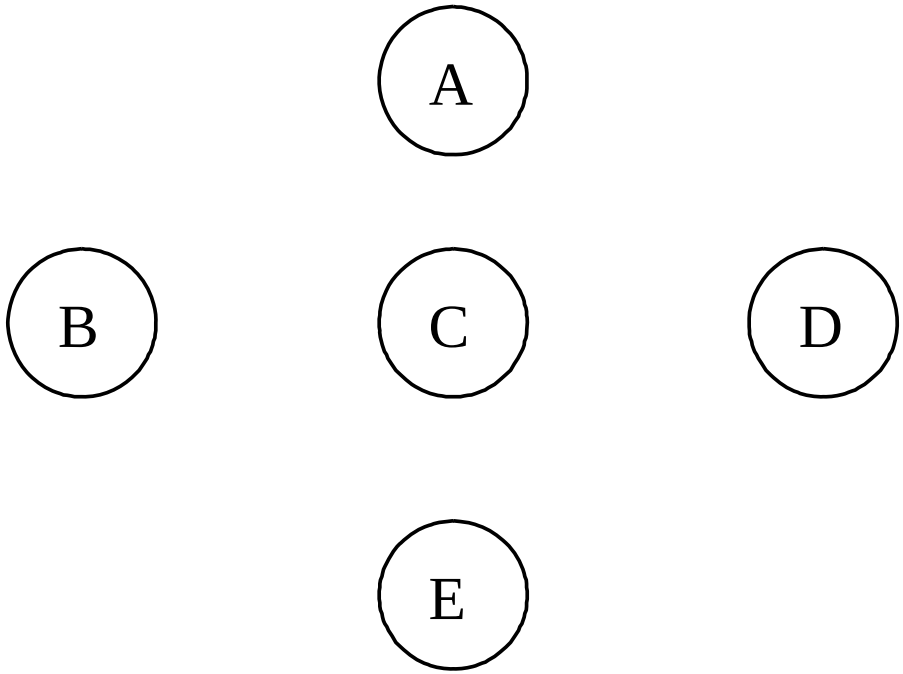
=

Cyber-Physical System with good performance

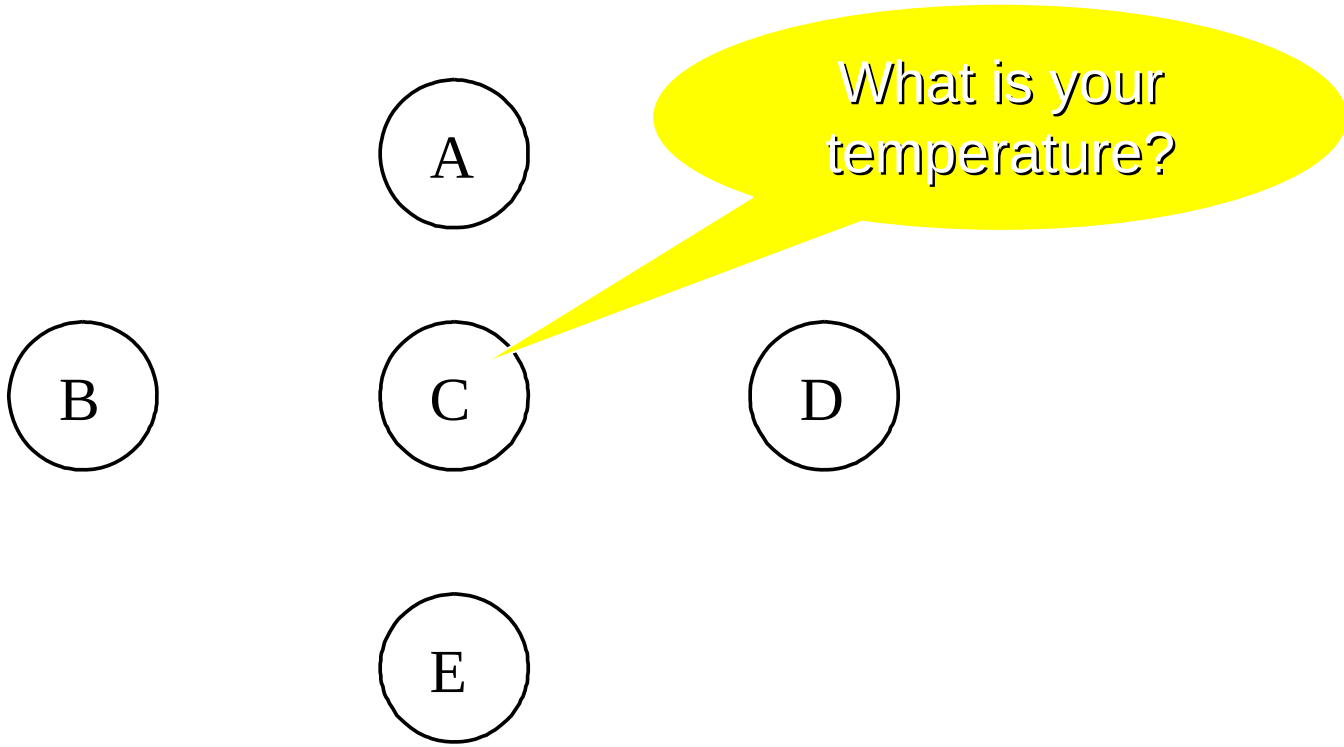
A Sensor Network



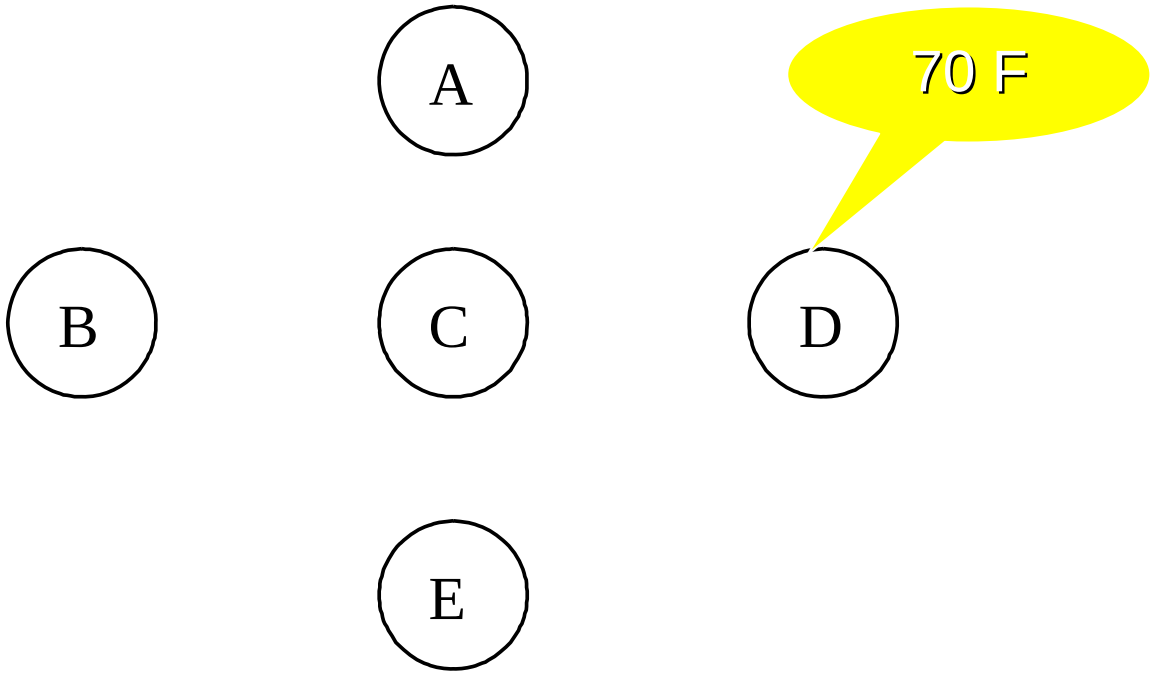
A Sensor Network



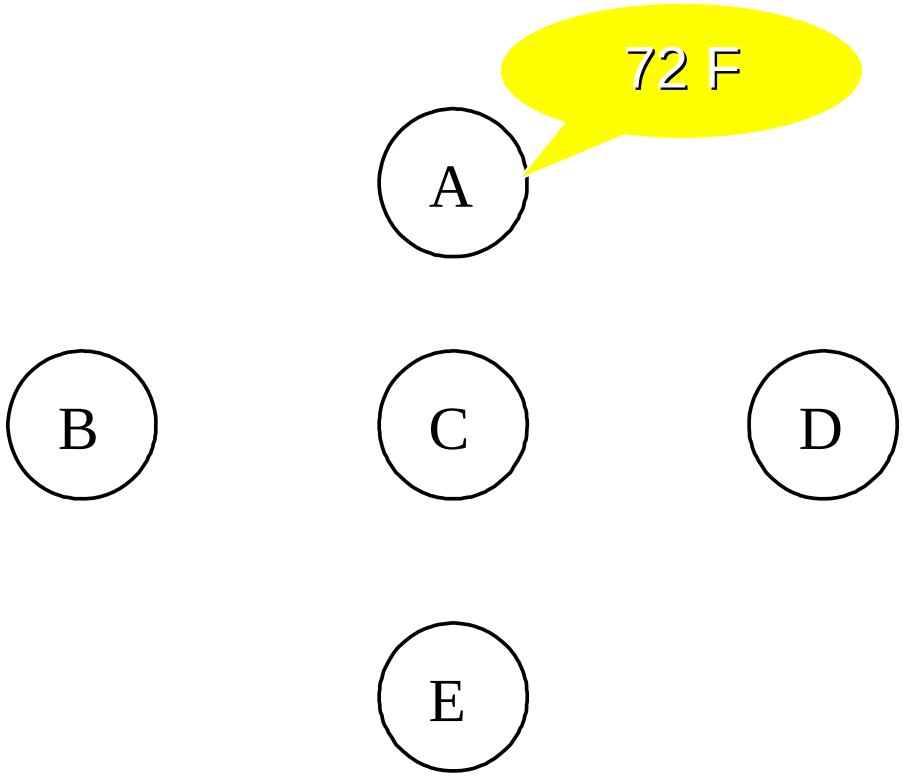
One node (node C) wants to know the minimum temperature in an area.



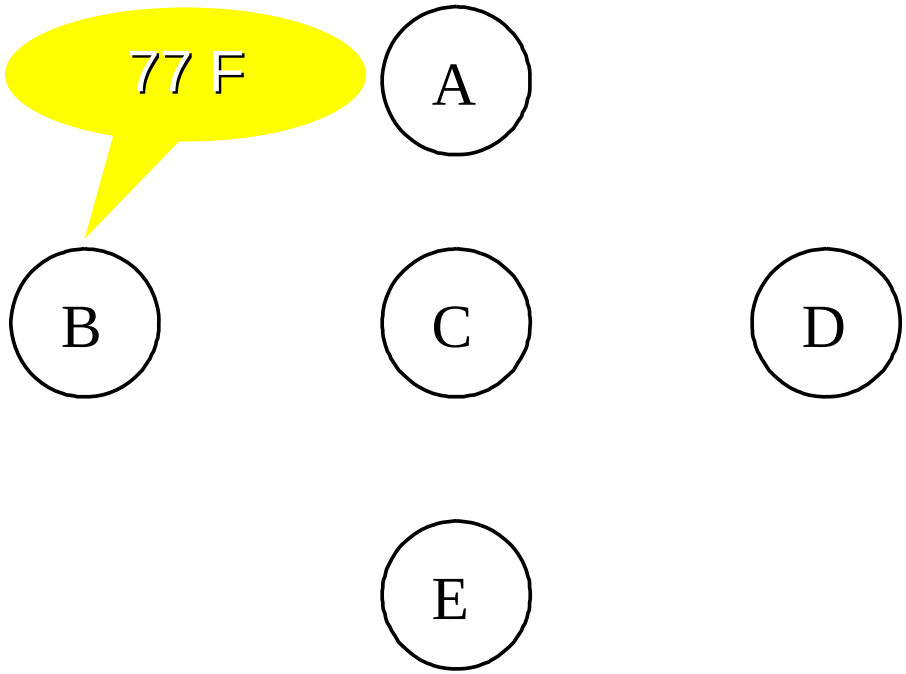
Node C broadcasts a request.



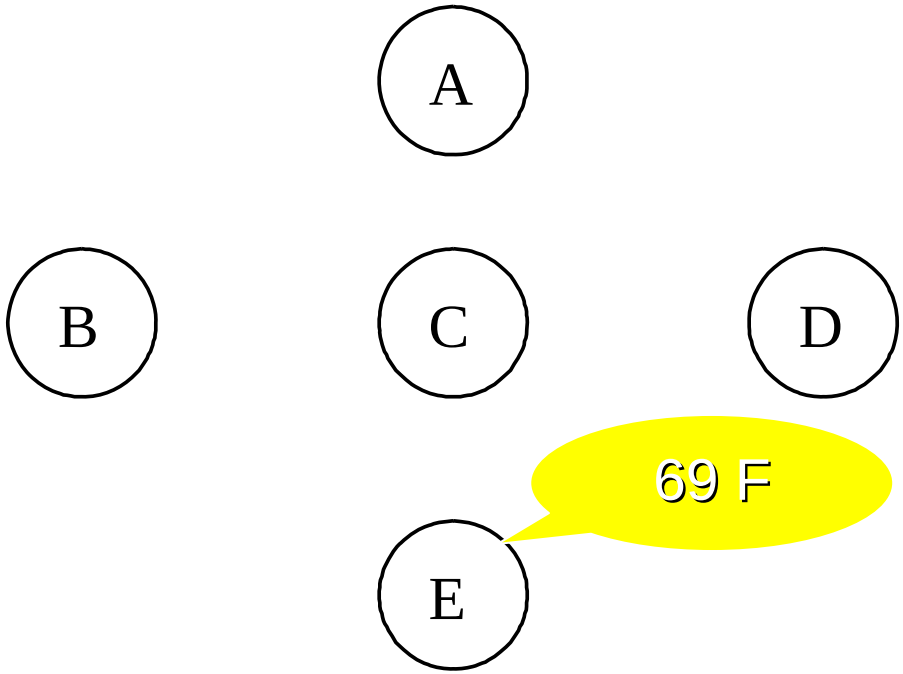
The other nodes respond.



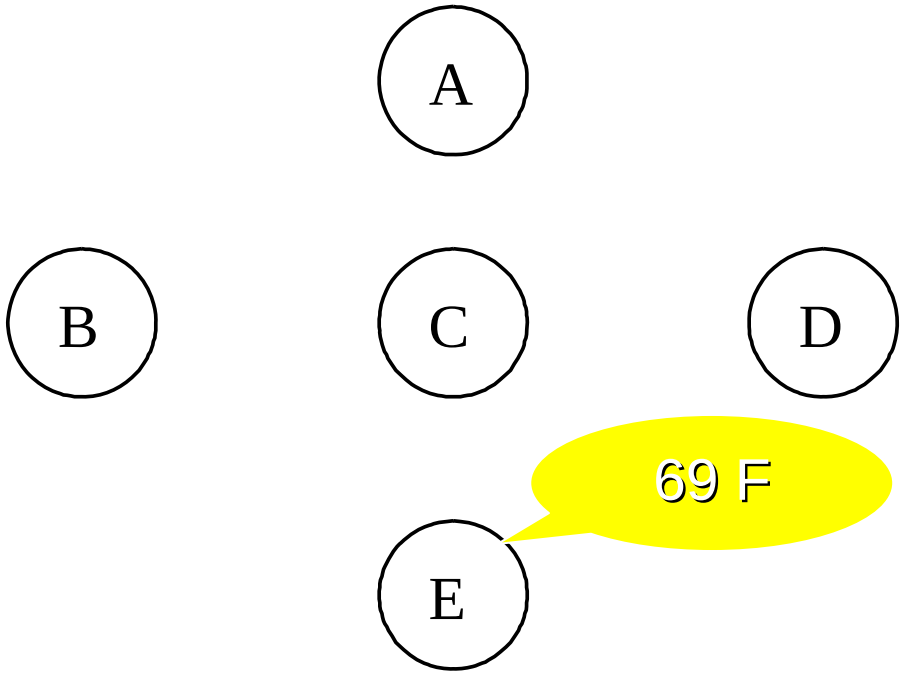
The other nodes respond.



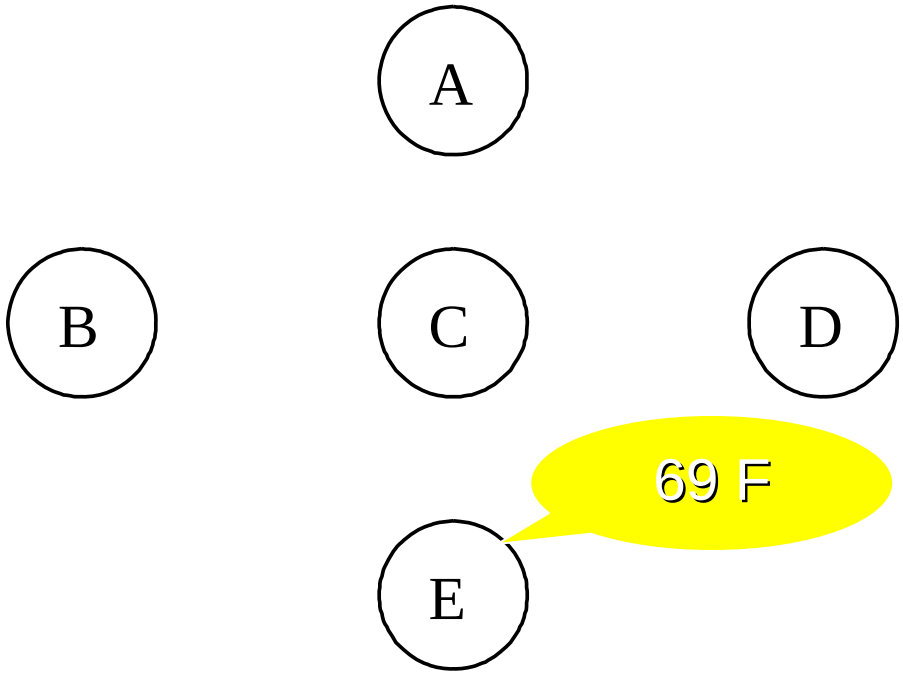
The other nodes respond.



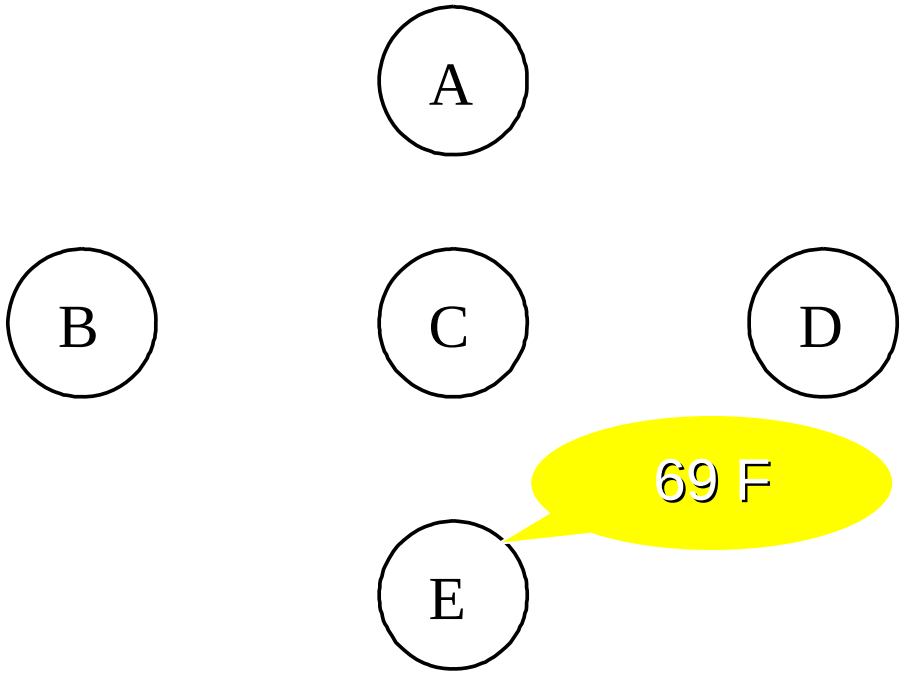
The other nodes respond.



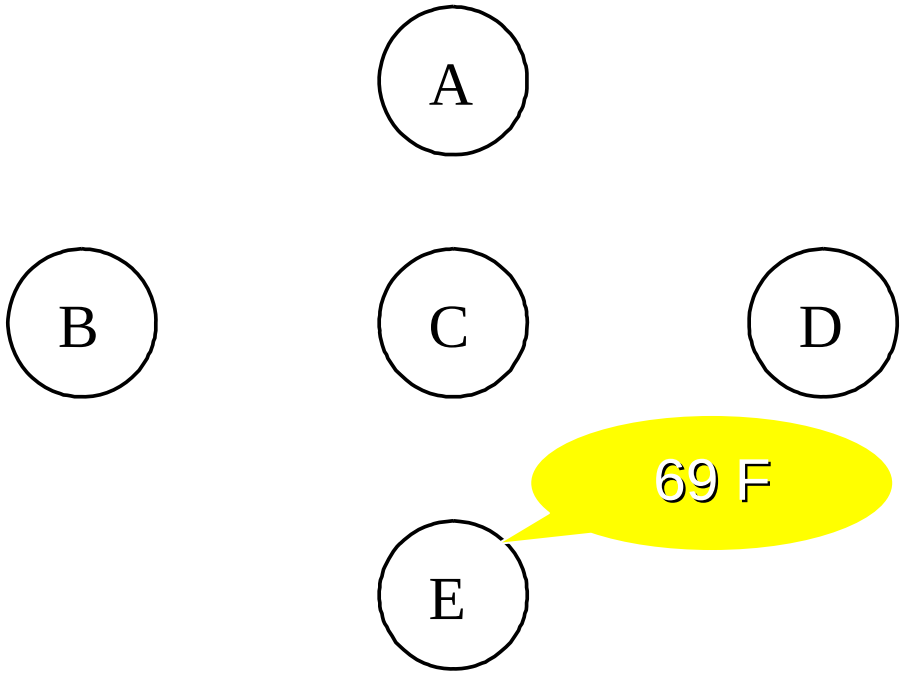
Now node C knows that min temperature is 69 F.



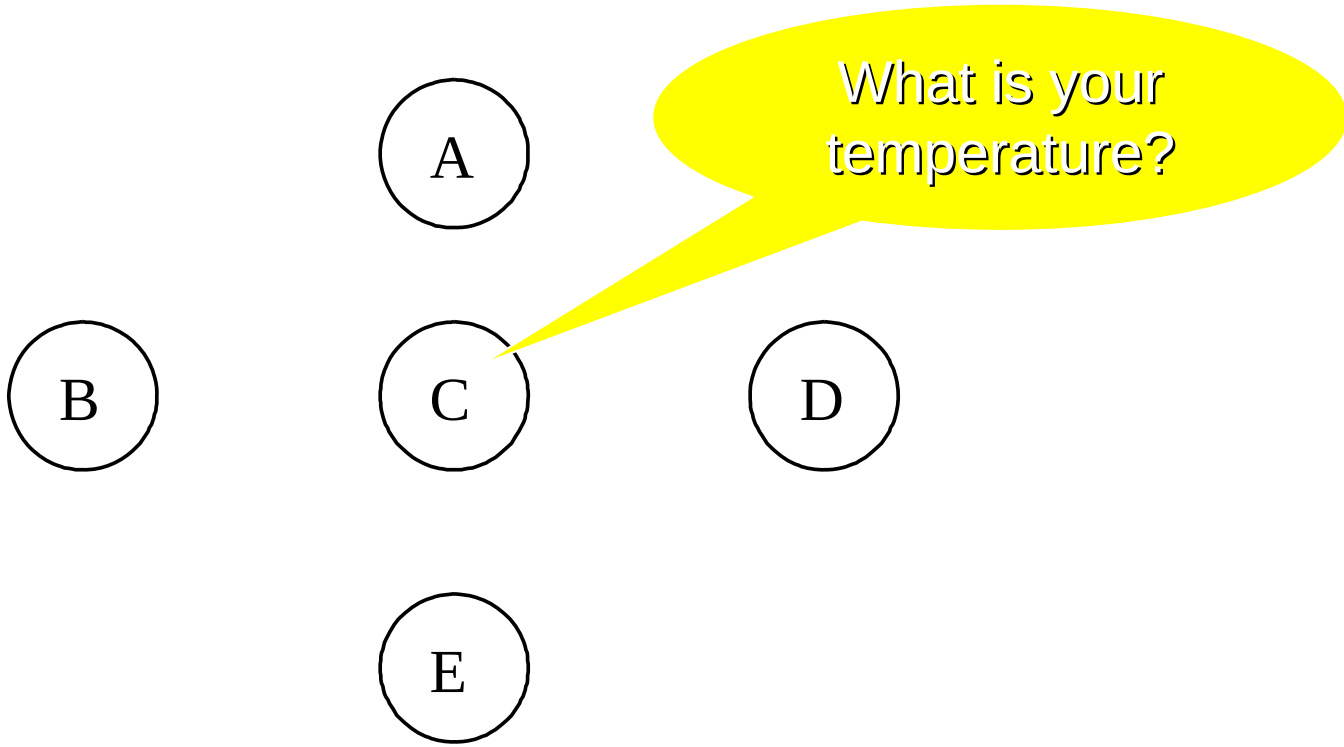
But it is slow…



With *m* nodes it takes O(*m*) time.

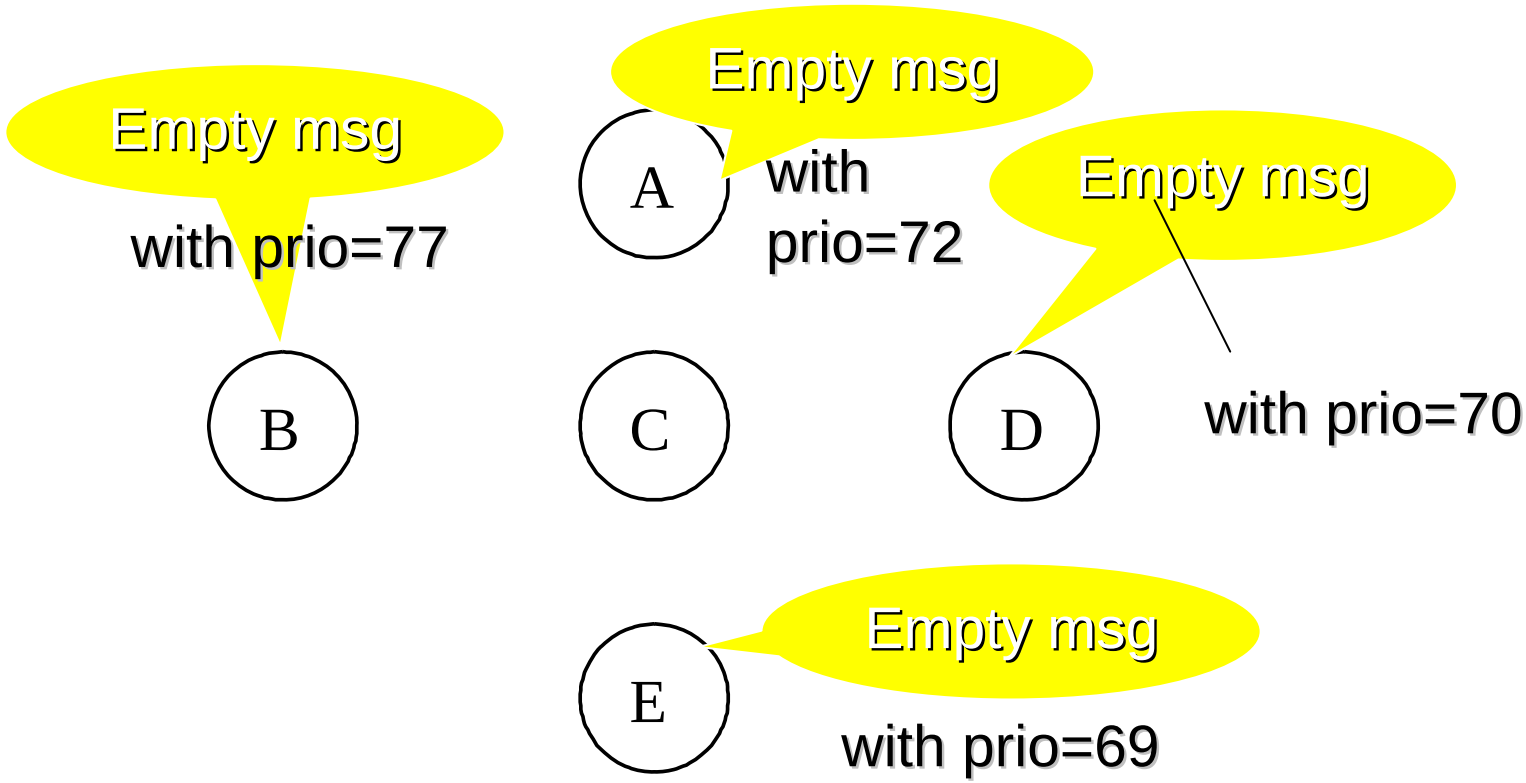


With a prioritized protocol, we can do it faster...



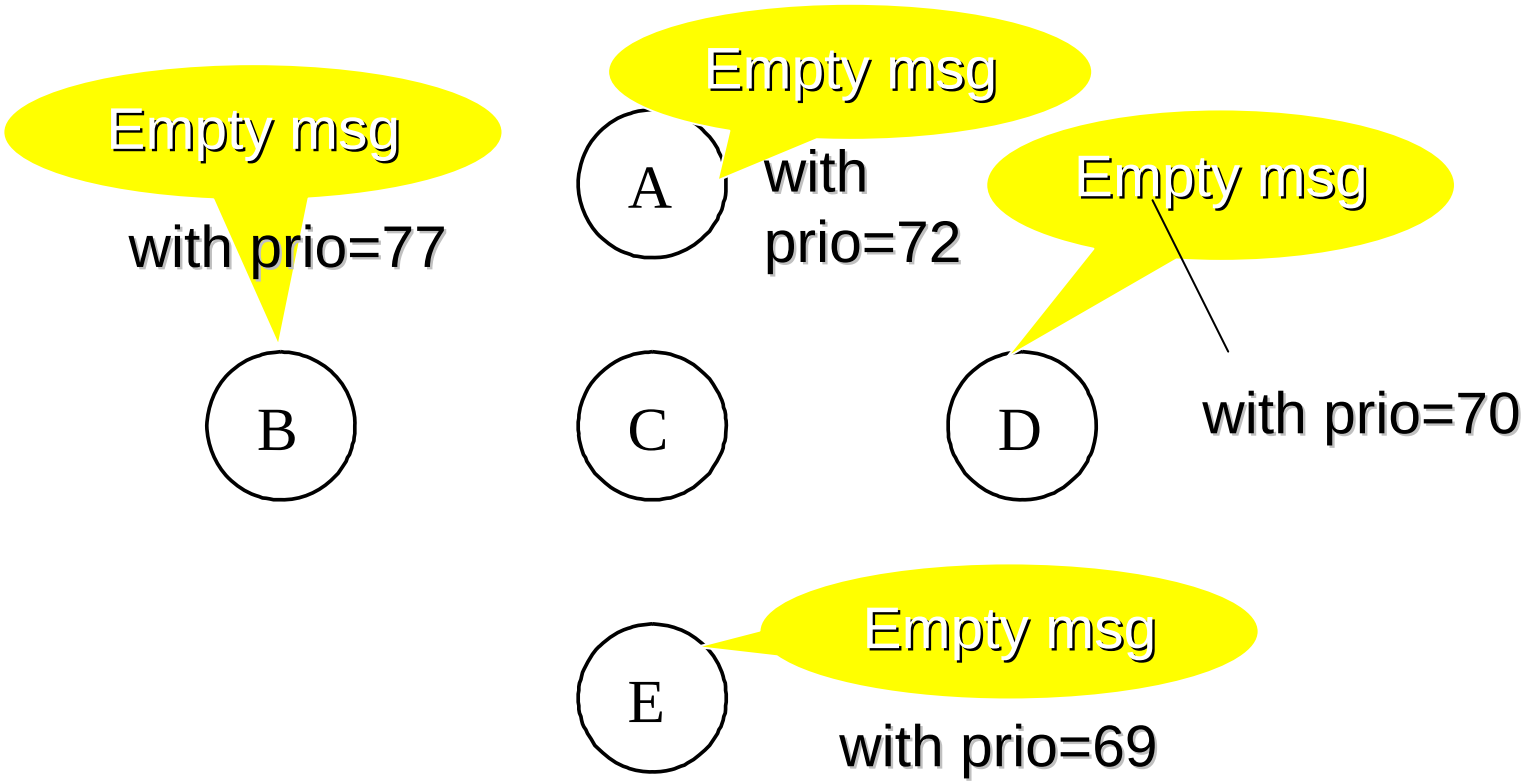
Node C broadcasts a request.

Computing Aggregated Quantities Efficiently



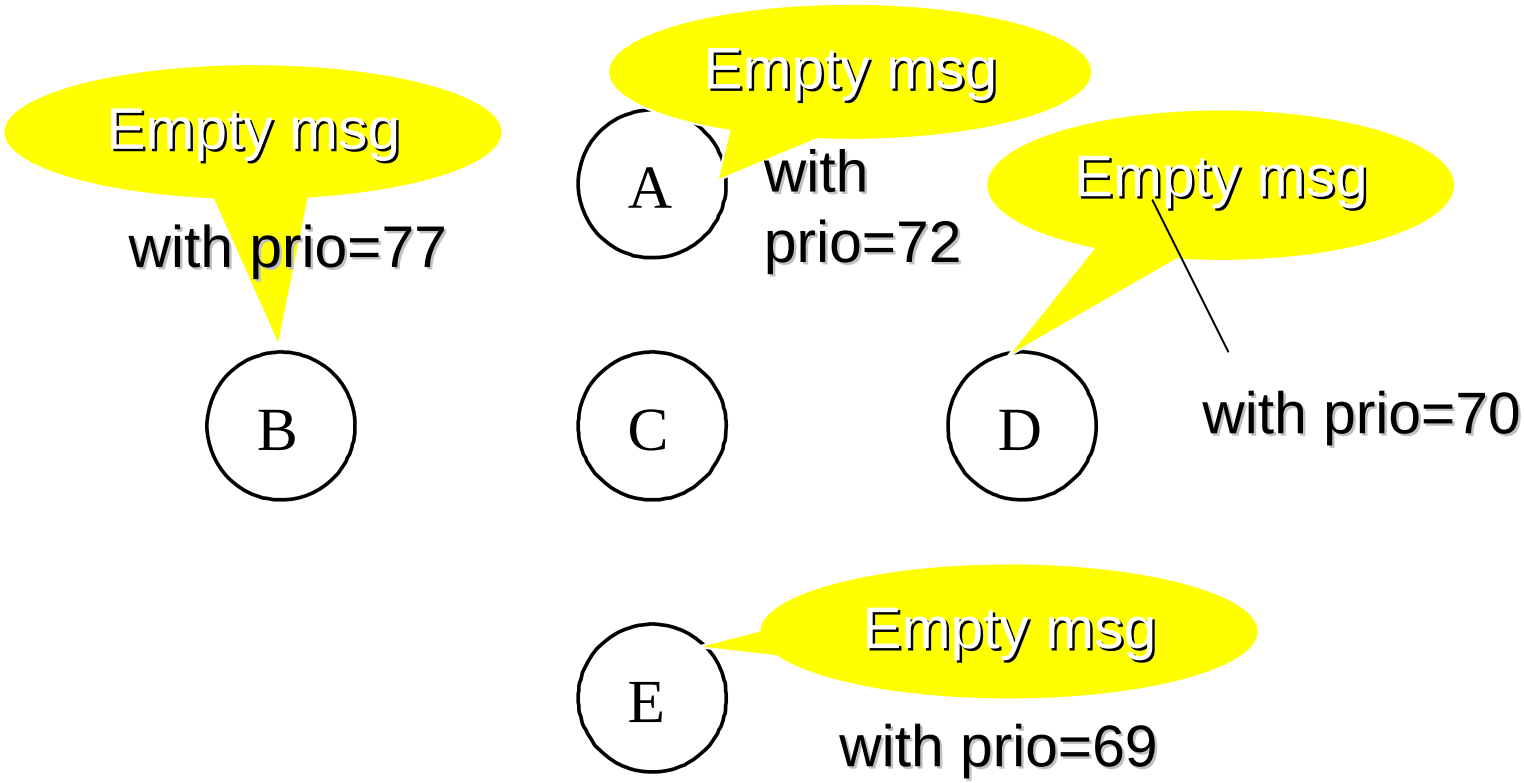
The other nodes respond in parallel…

Computing Aggregated Quantities Efficiently



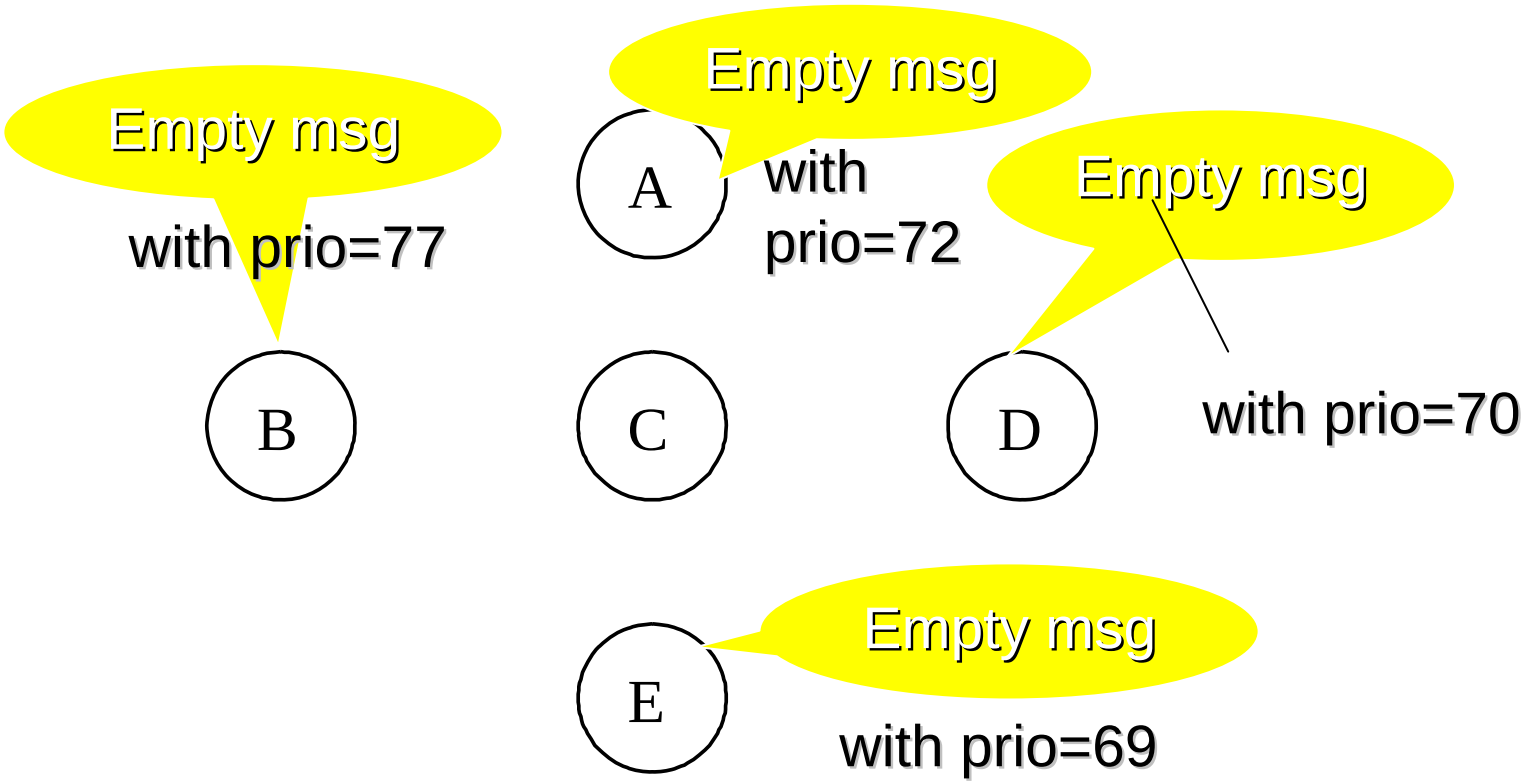
By transmitting empty msg. The priority of the msg is given by the temperature.

Computing Aggregated Quantities Efficiently



Now node C knows that min temperature is 69 F.

Computing Aggregated Quantities Efficiently



But this is faster. Time complexity in O(1).

More computations

√ Computing MIN

More computations

- √ Computing MIN
- Computing MAX
- Computing COUNT
- Computing MEDIAN
- Computing Interpolation of sensor data

More computations

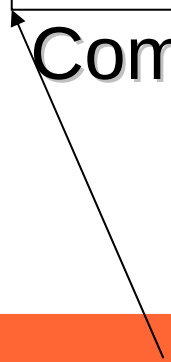
√ Computing MIN

Computing MAX

Computing COUNT

Computing MEDIAN

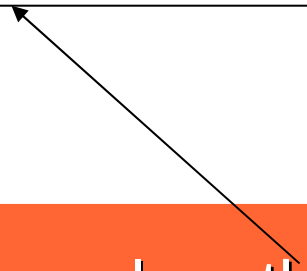
Computing Interpolation of sensor data



These can be computed efficiently as well.

More computations

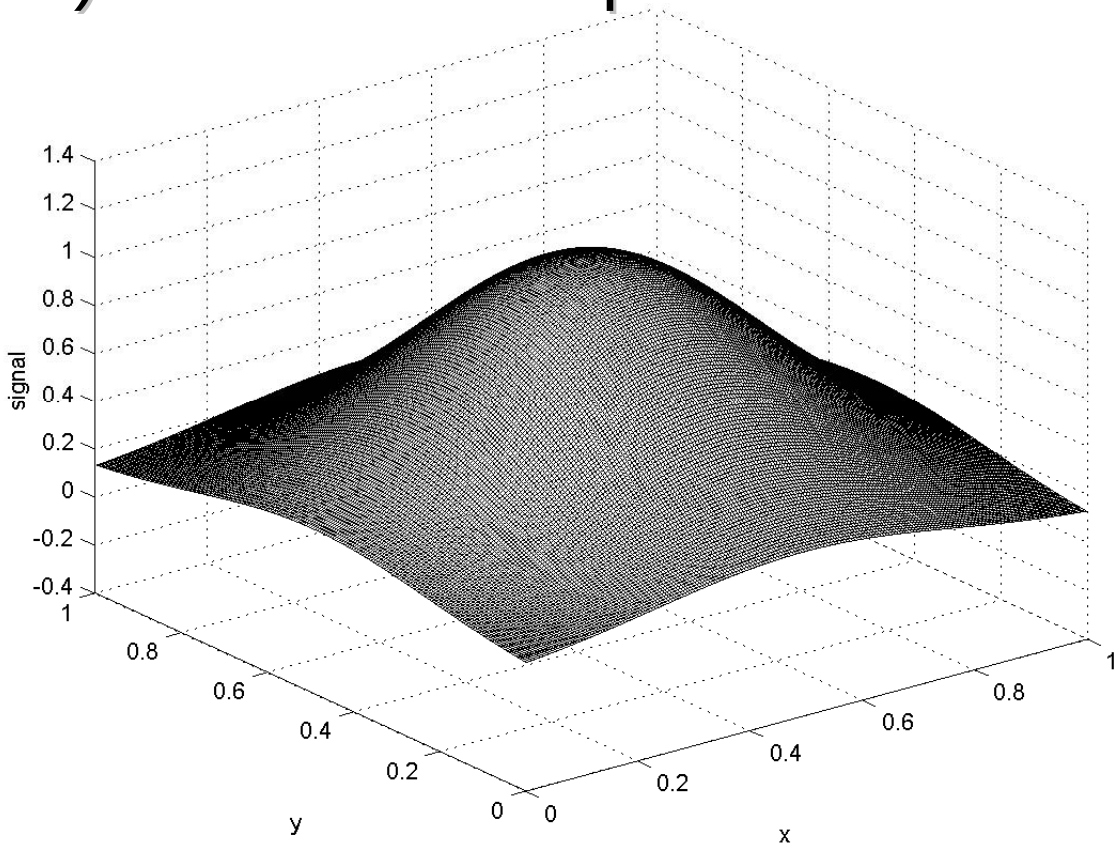
- √ Computing MIN
- Computing MAX
- Computing COUNT
- Computing MEDIAN
- Computing Interpolation of sensor data



Let us explore this.

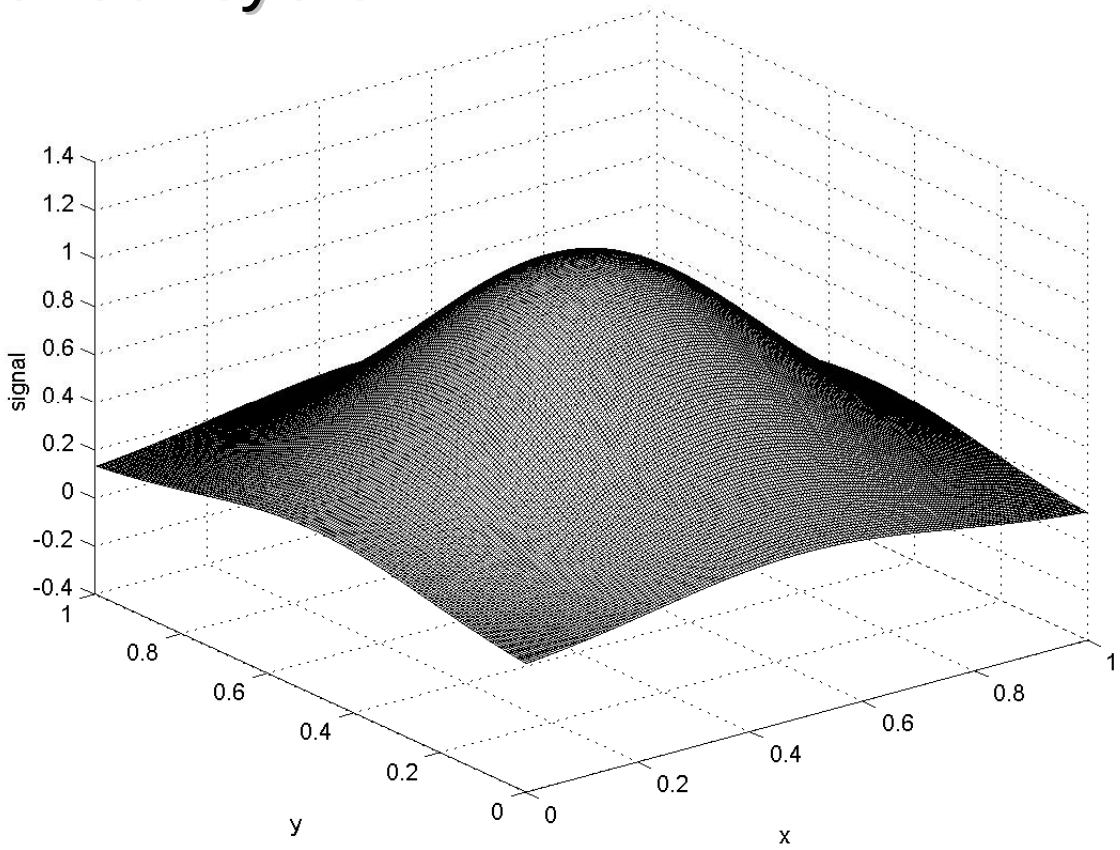
Computing Interpolation

Let us consider a signal (say concentration of a hazardous gas) that varies in space.



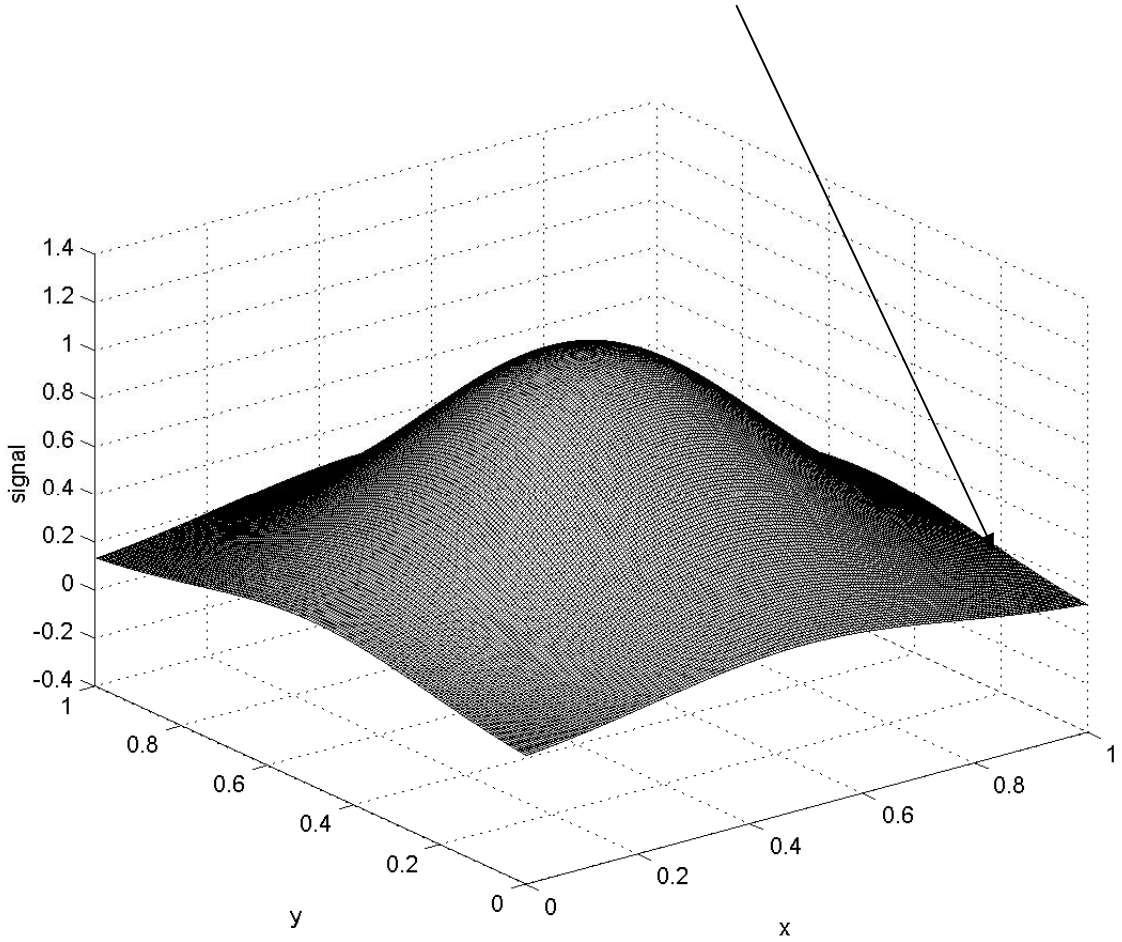
Computing Interpolation

How can we get this signal from a networked embedded sensor system?



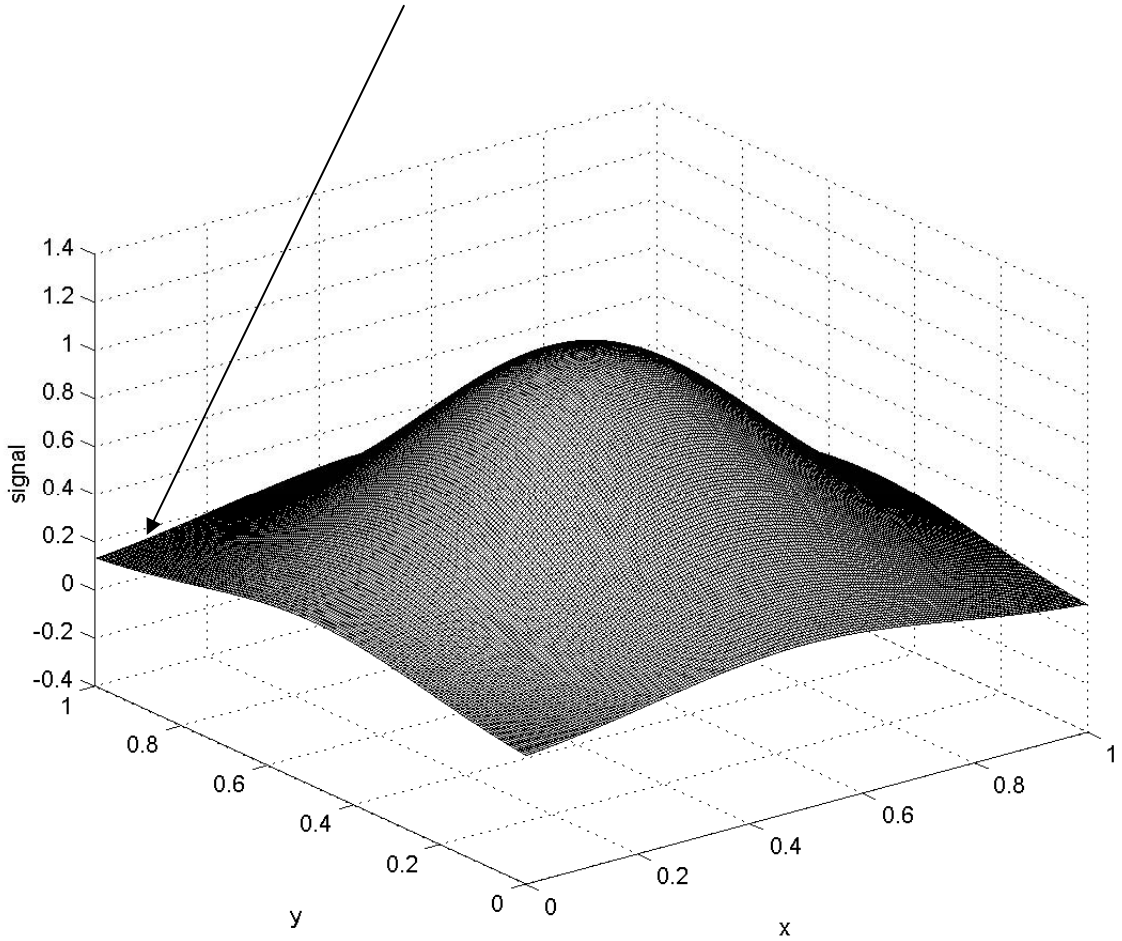
Computing Interpolation

We can deploy one sensor node here…



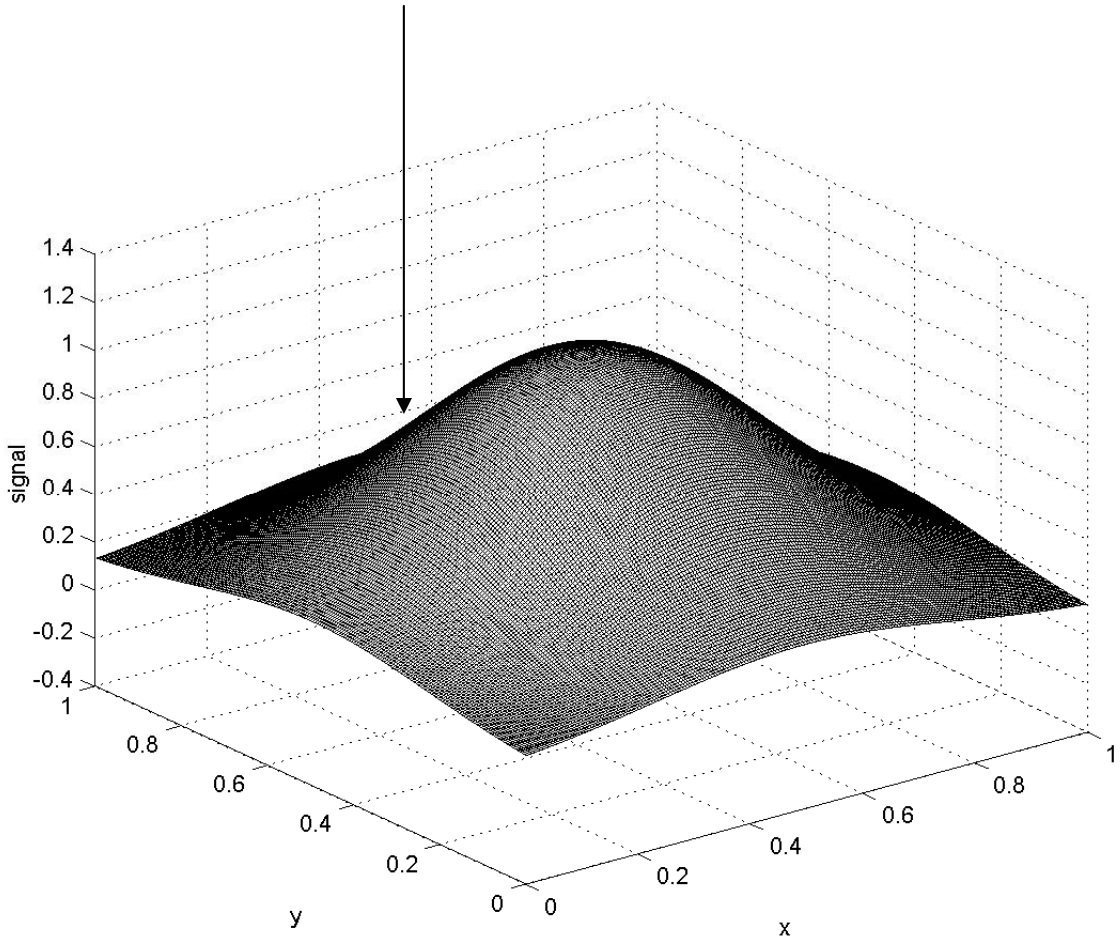
Computing Interpolation

… and another one here….



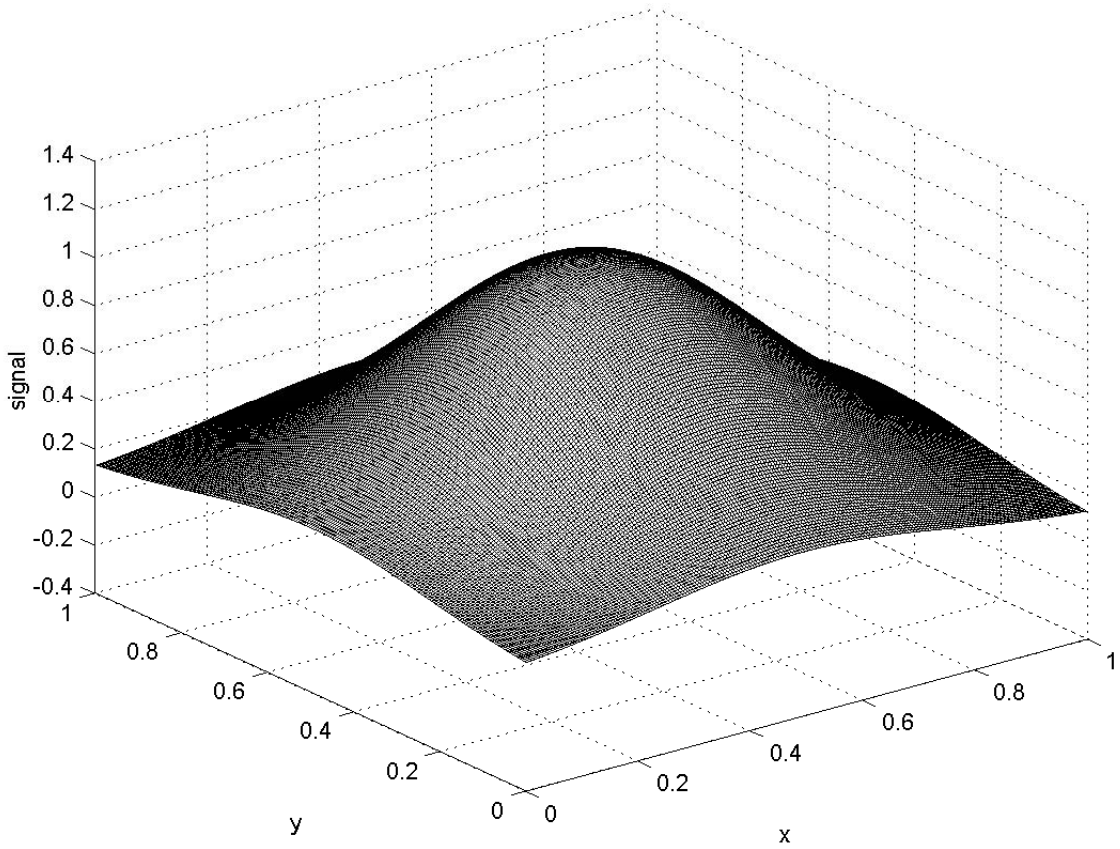
Computing Interpolation

… and another one here….



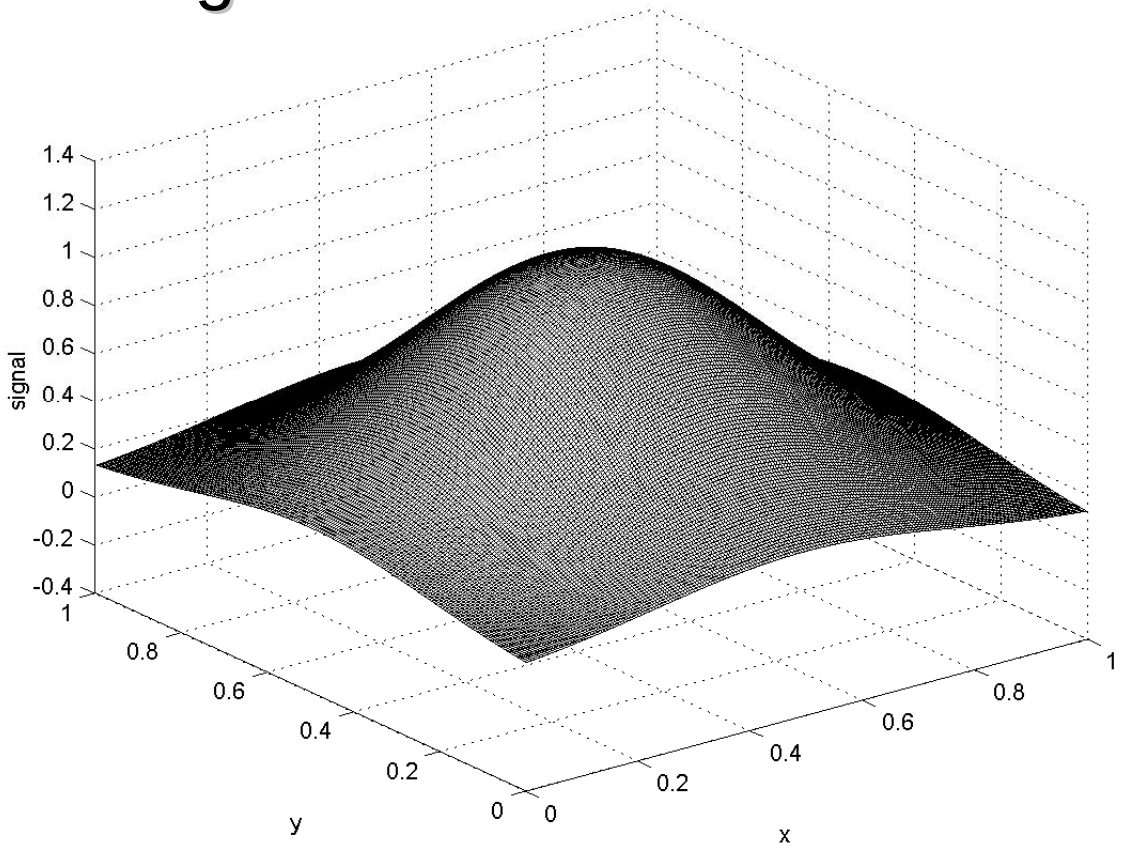
Computing Interpolation

… until we have a dense network.



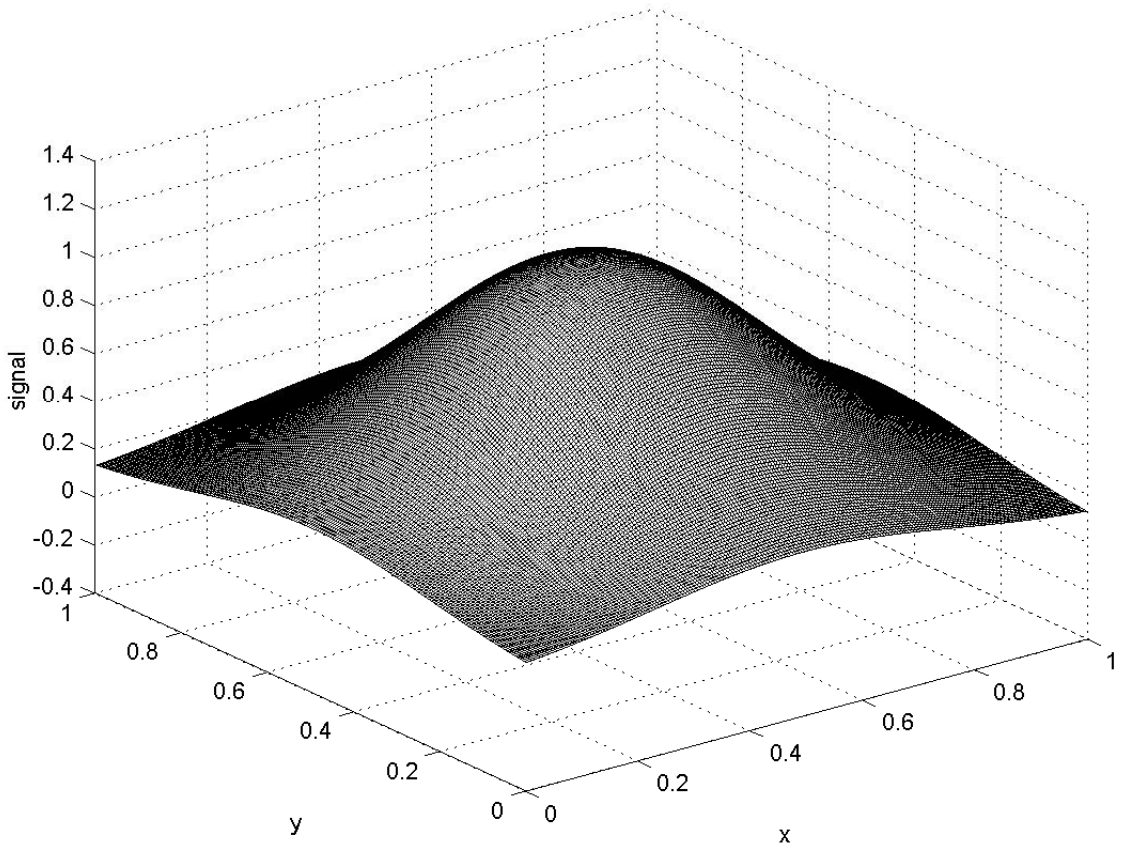
Computing Interpolation

When the signal changes with time we want to know the shape of the signal and how it moves?



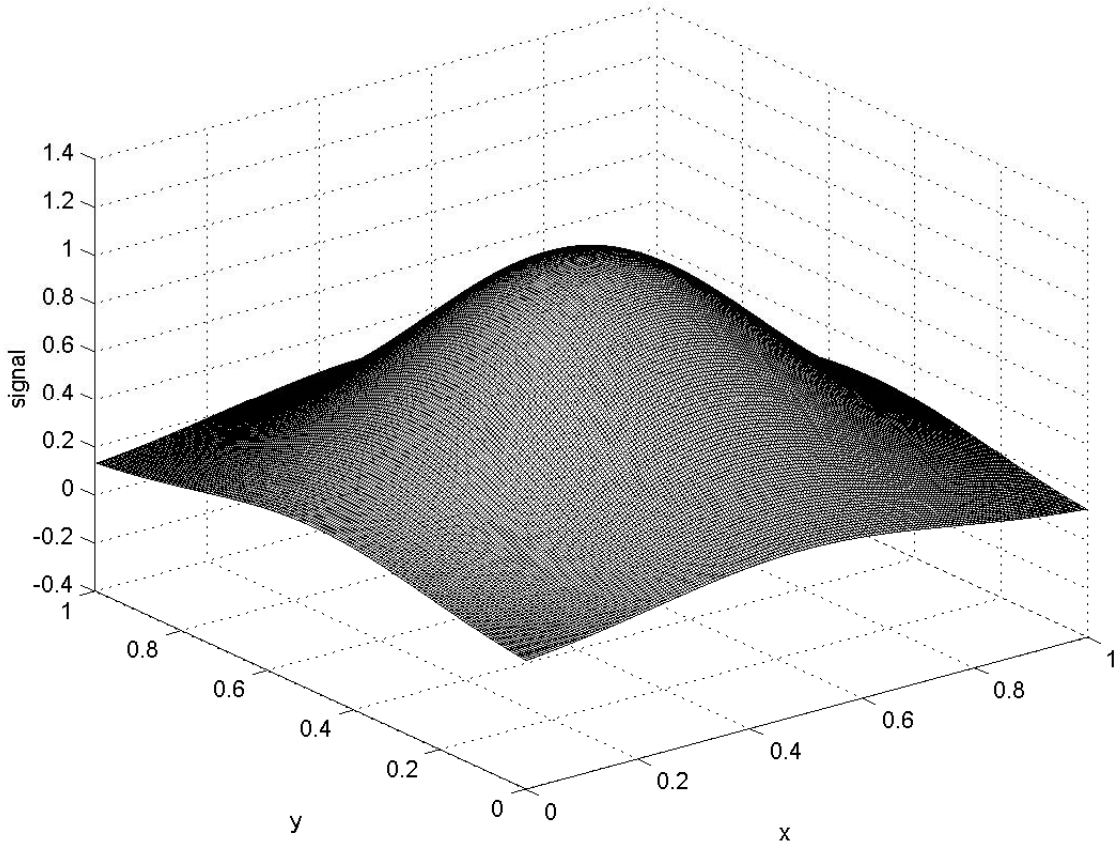
Computing Interpolation

And we want to do that quickly (say a few times per s).



Computing Interpolation

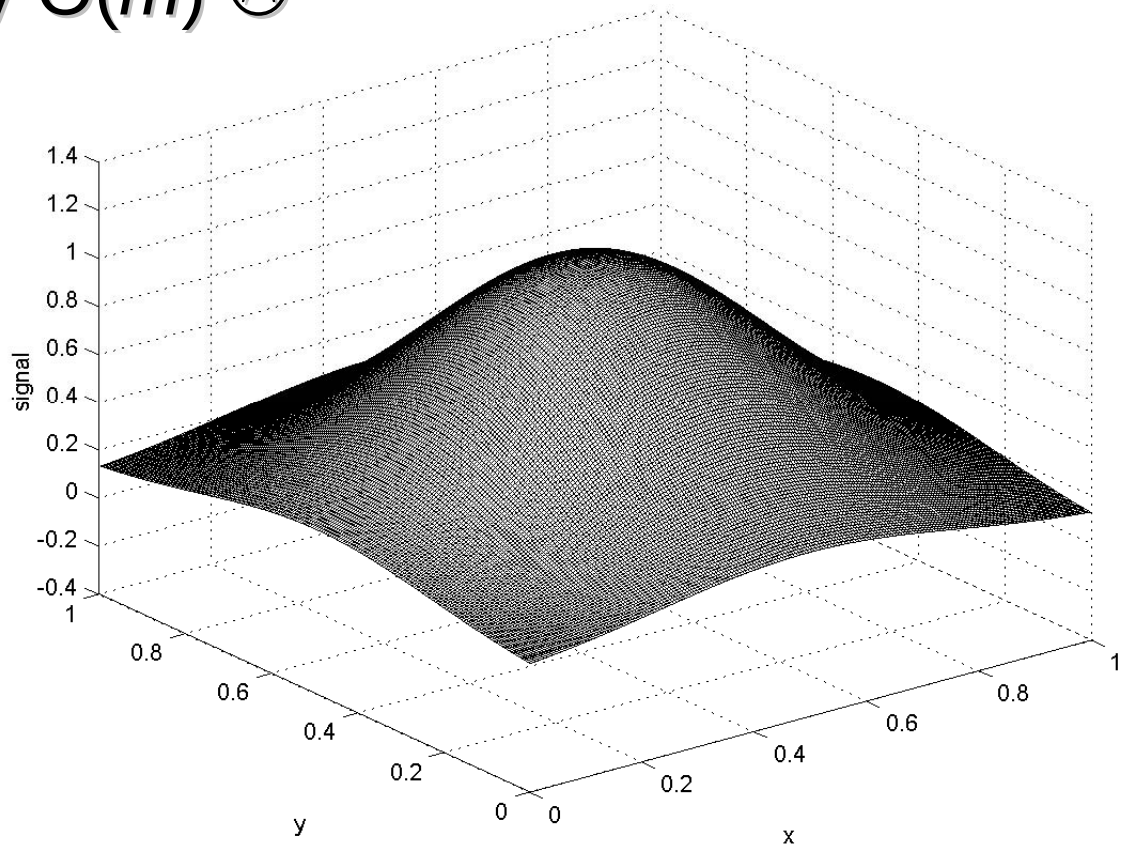
Naïve solution: Let each node broadcast its sensor value…



Computing Interpolation

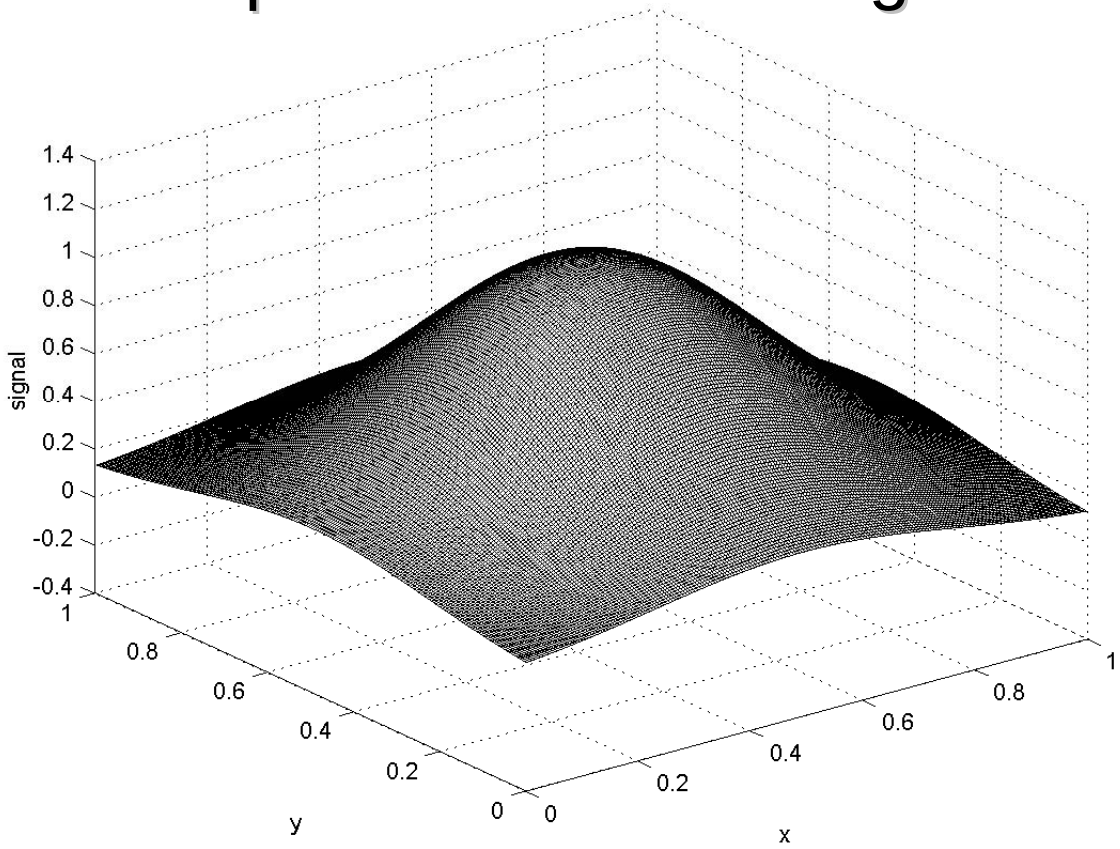
… and then perform curve-fitting.

Time complexity O(*m*) ☹



Computing Interpolation

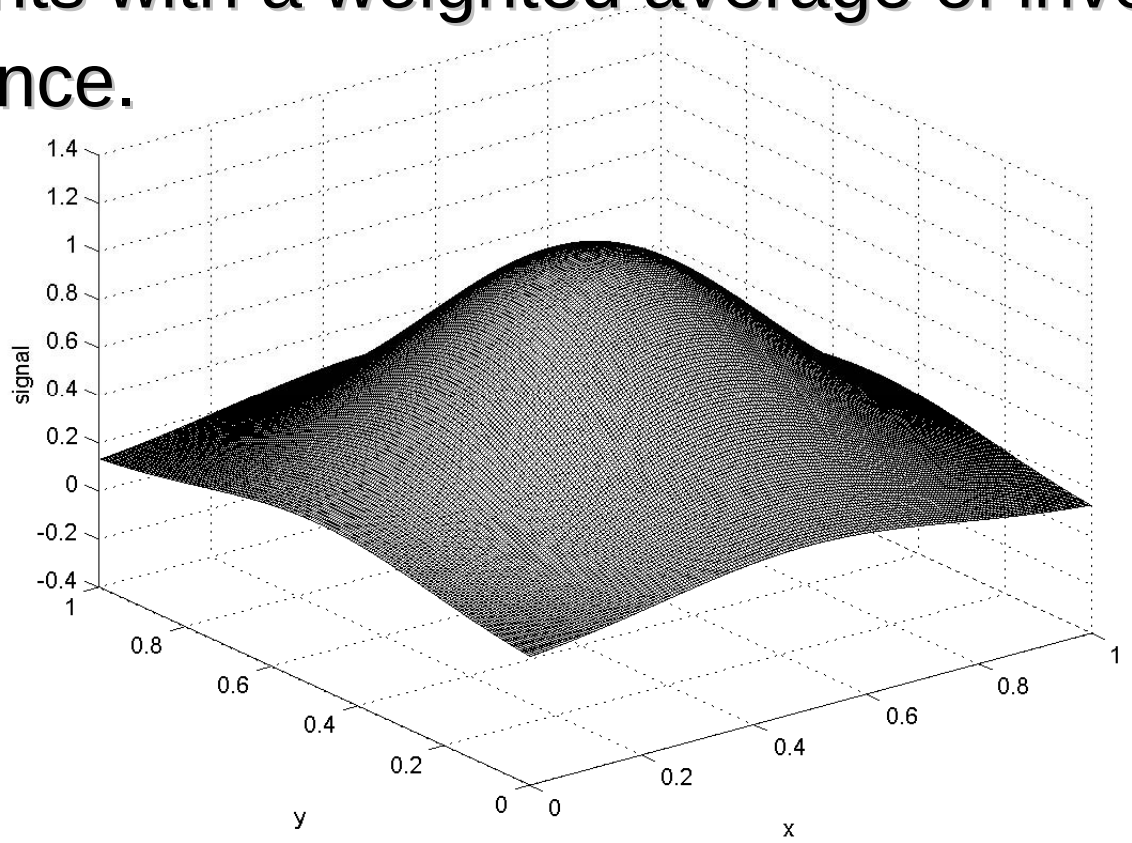
Slightly intelligent solution: Let a subset of nodes broadcast and then perform curve-fitting.



Computing Interpolation

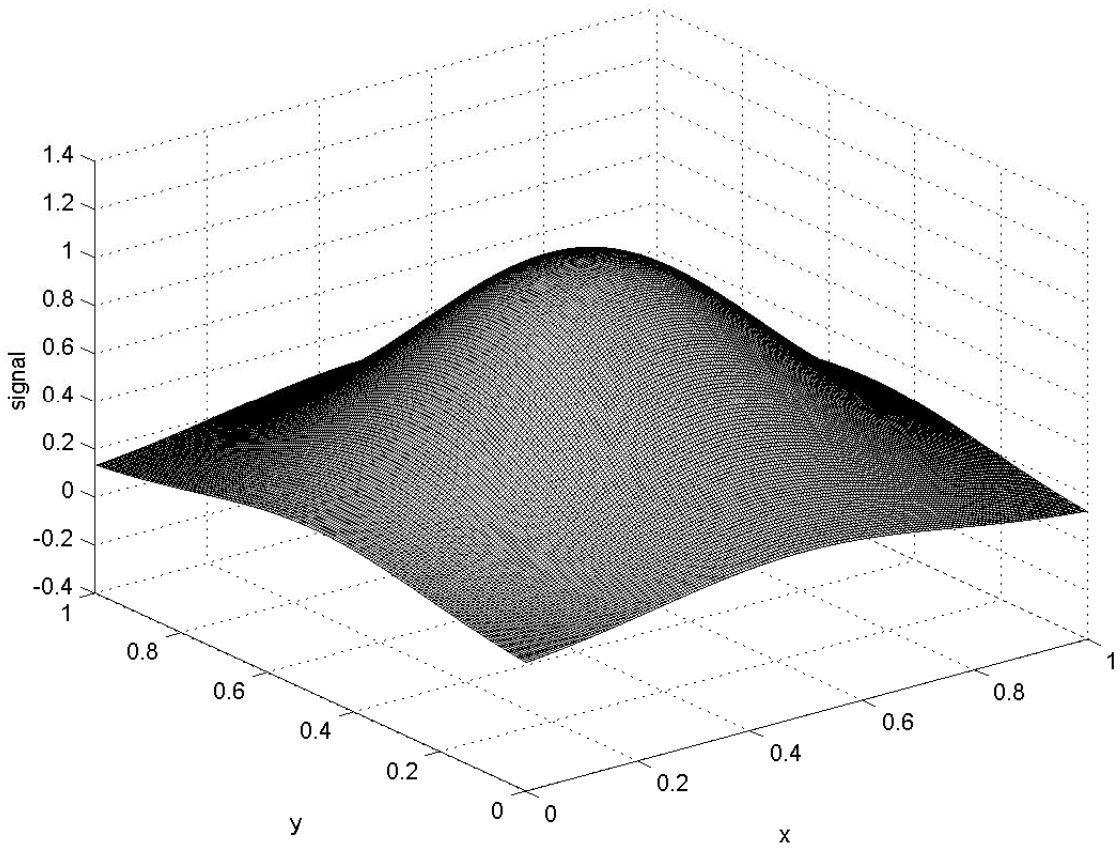
Q: And how to perform curve-fitting?

A: Use data points with a weighted average of inverse square distance.



Computing Interpolation

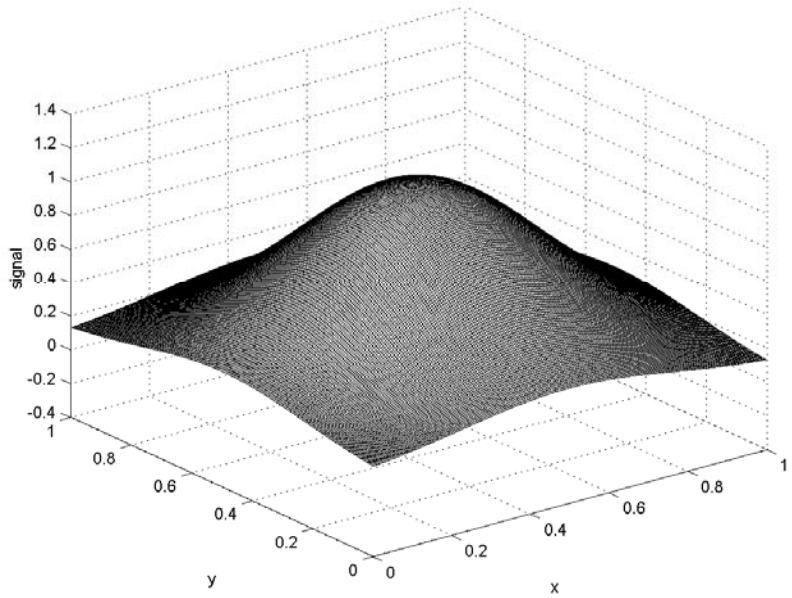
Q: But which nodes should be selected?



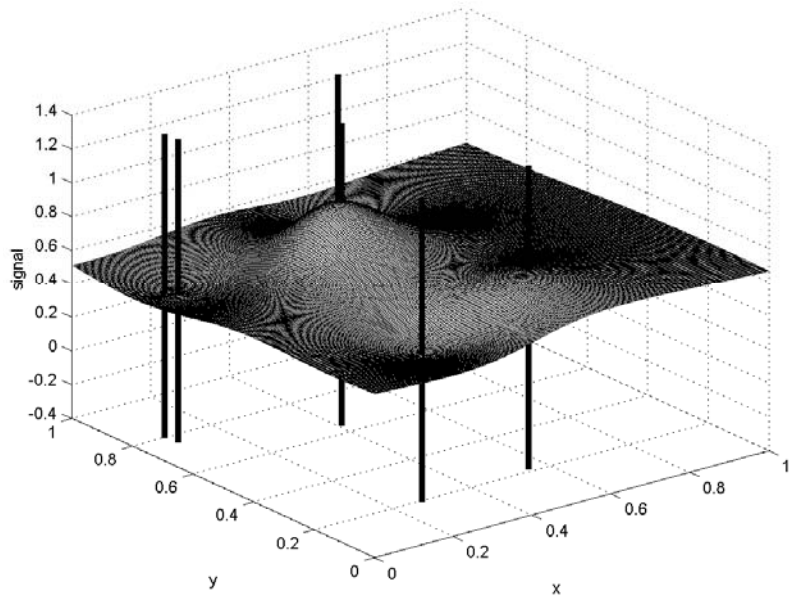
Computing Interpolation

Select nodes at random.

Signal



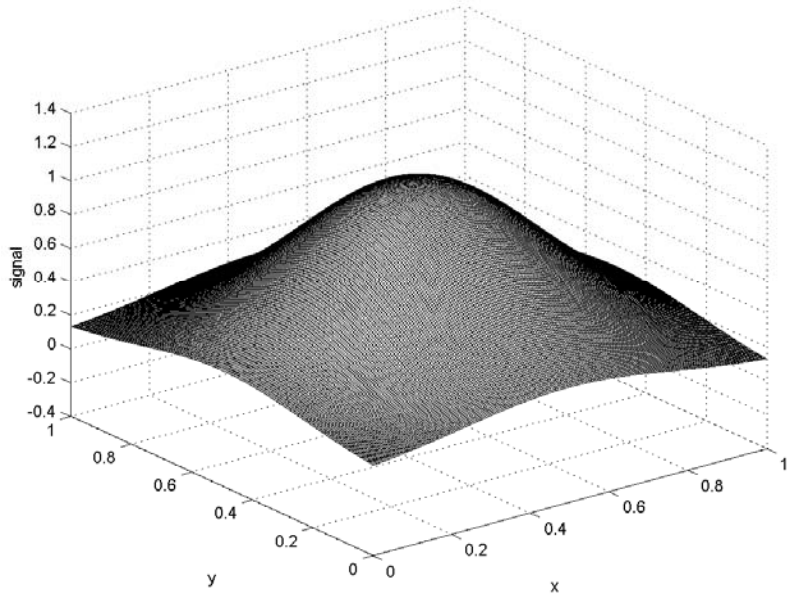
Curve fitting



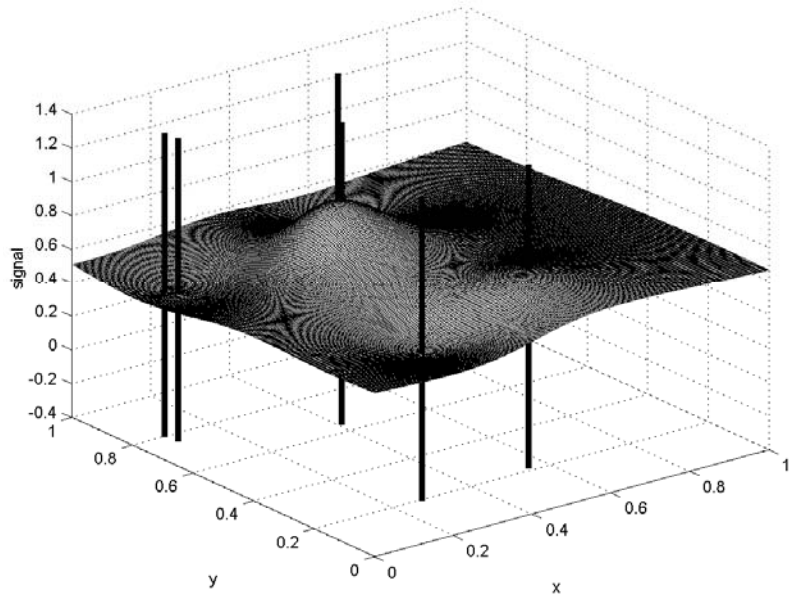
Computing Interpolation

It gives a poor representation of the data.

Signal



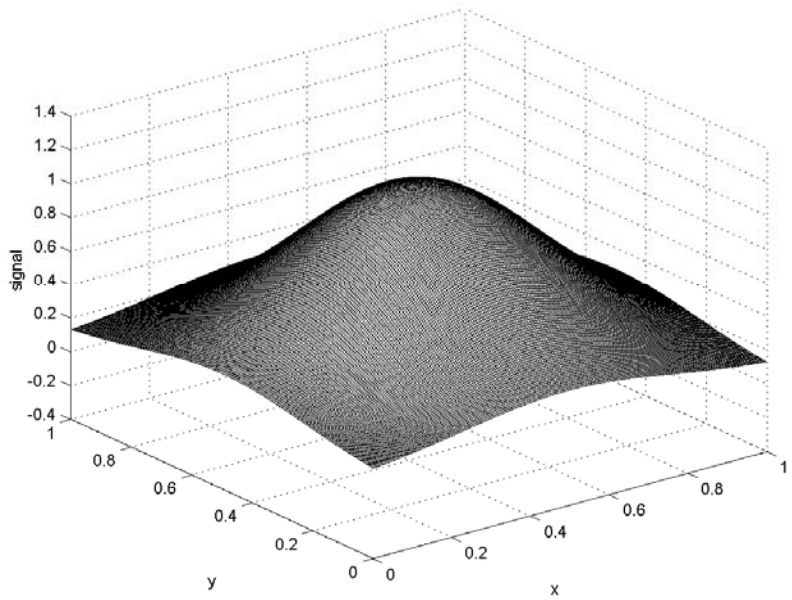
Curve fitting



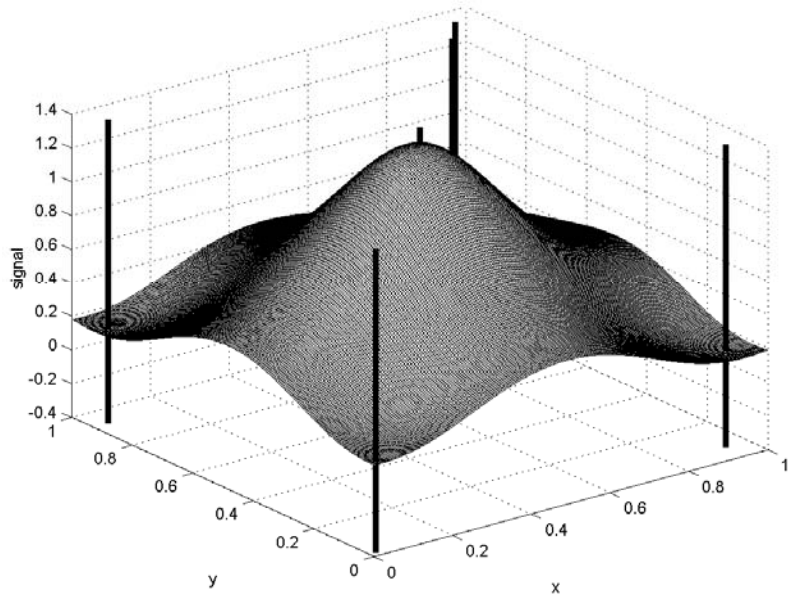
Computing Interpolation

Let us select some other points.

Signal



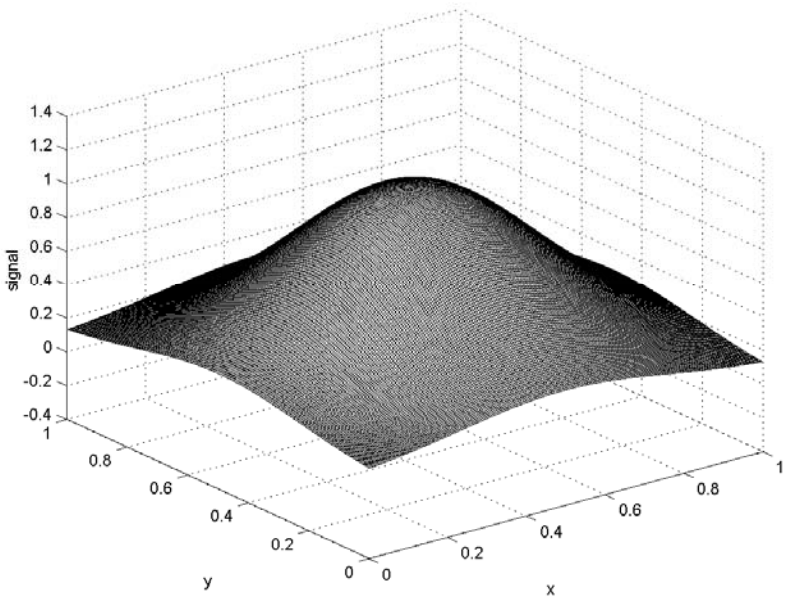
Curve fitting



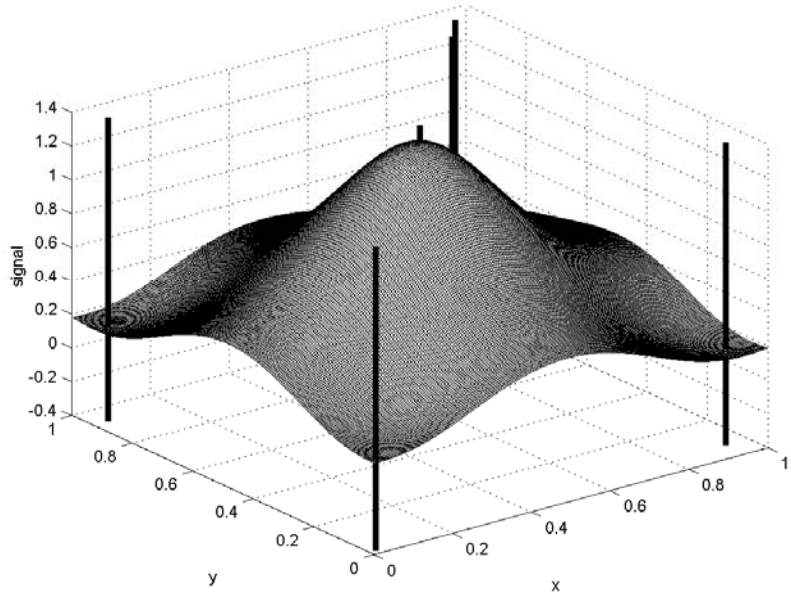
Computing Interpolation

And we get a good fit ☺

Signal



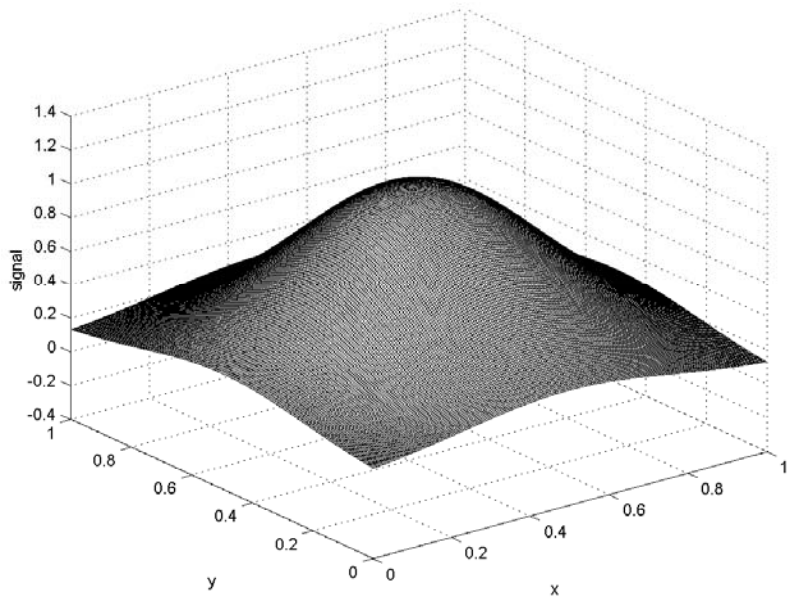
Curve fitting



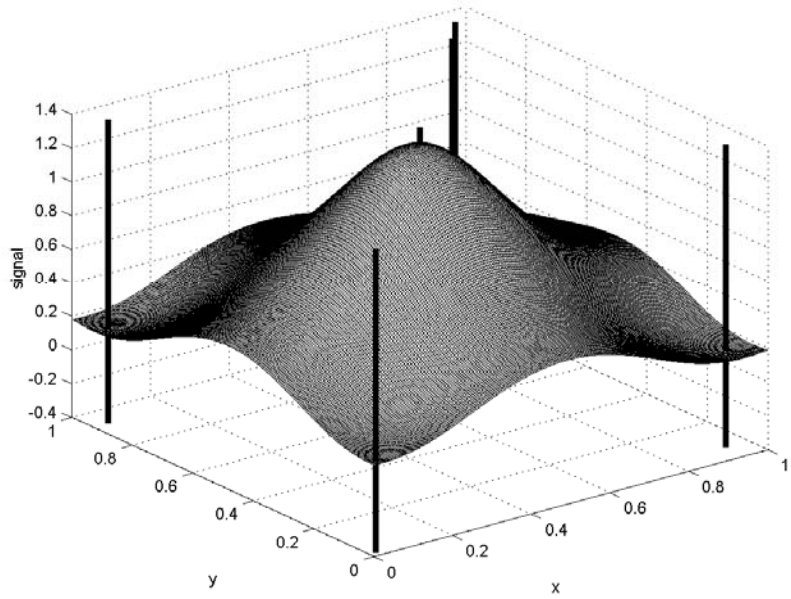
Computing Interpolation

Curve fitting is good if we can find the “right “data points.

Signal



Curve fitting

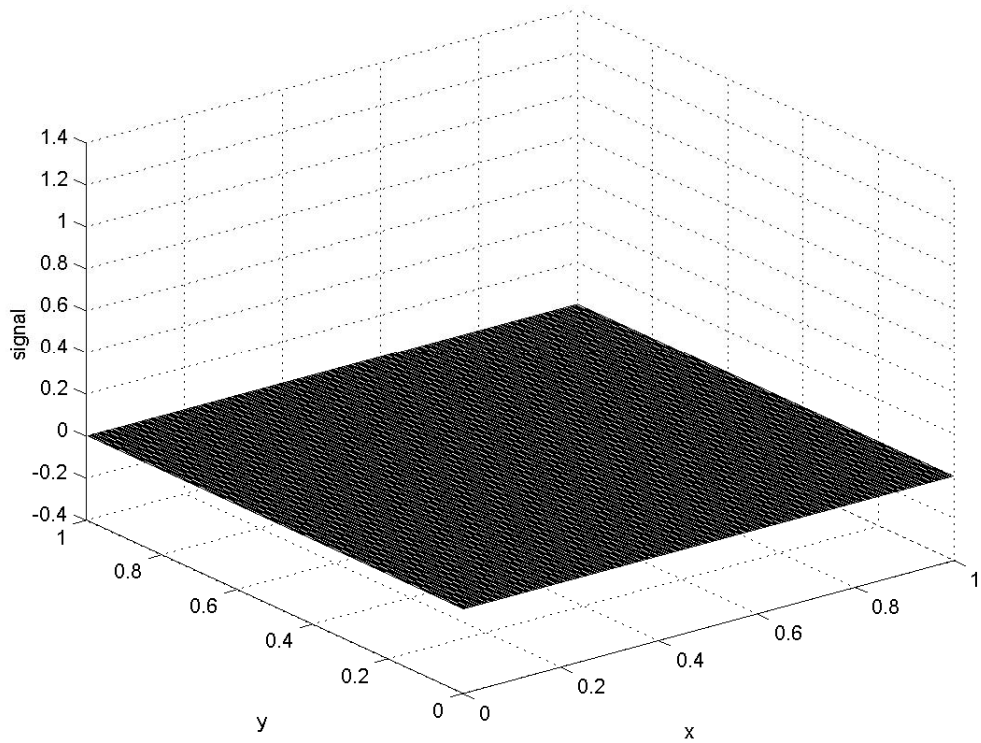


Computing Interpolation

Q: Which nodes should be selected?

A: Select the node with the largest error.

No data points
At all.

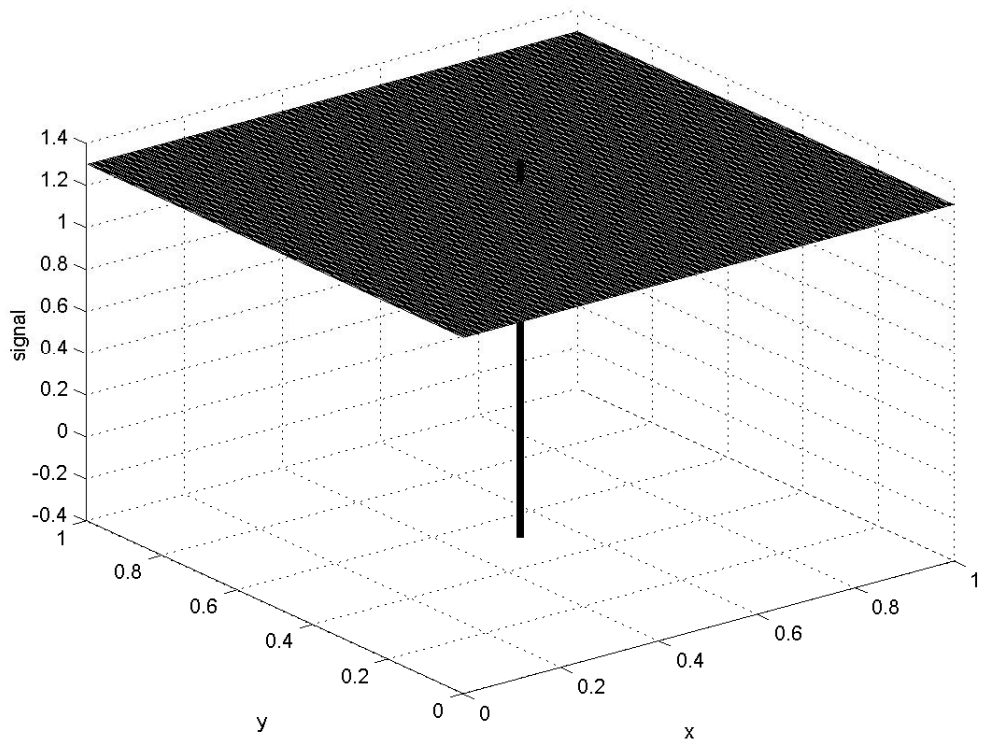


Computing Interpolation

Q: Which nodes should be selected?

A: Select the node with the largest error.

1 data point.

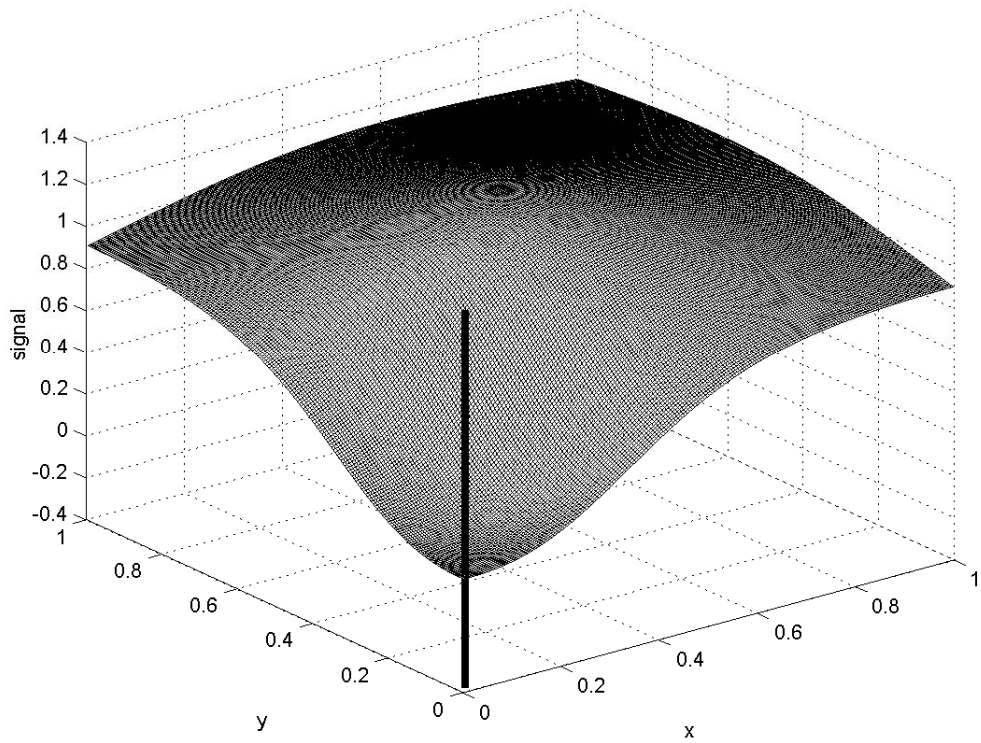


Computing Interpolation

Q: Which nodes should be selected?

A: Select the node with the largest error.

2 data points.

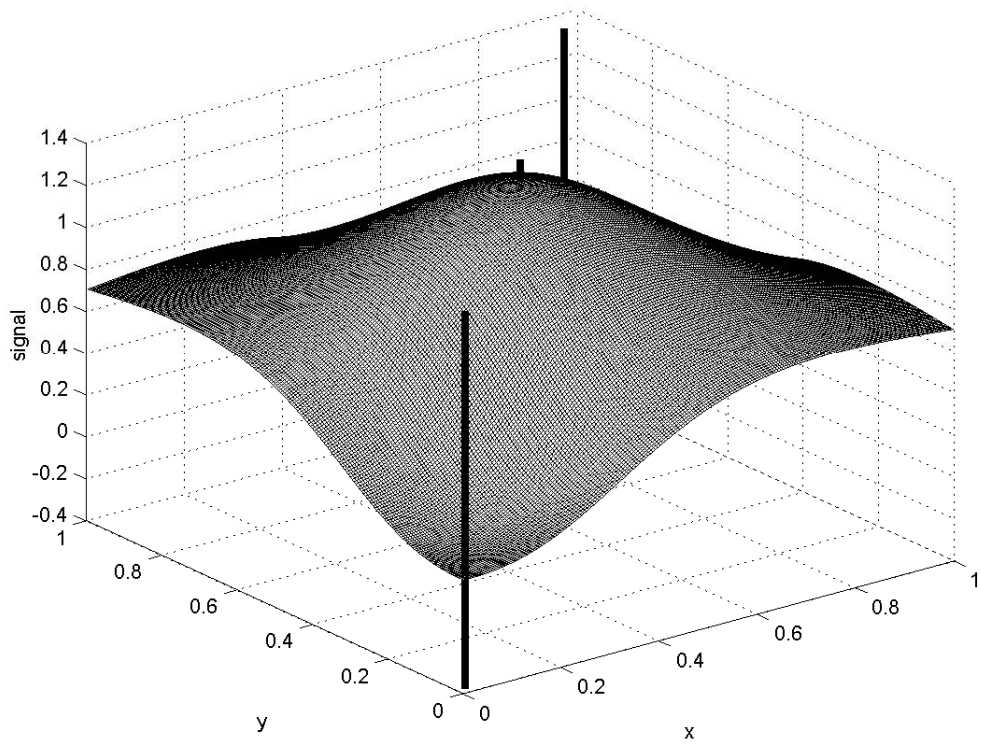


Computing Interpolation

Q: Which nodes should be selected?

A: Select the node with the largest error.

3 data points.

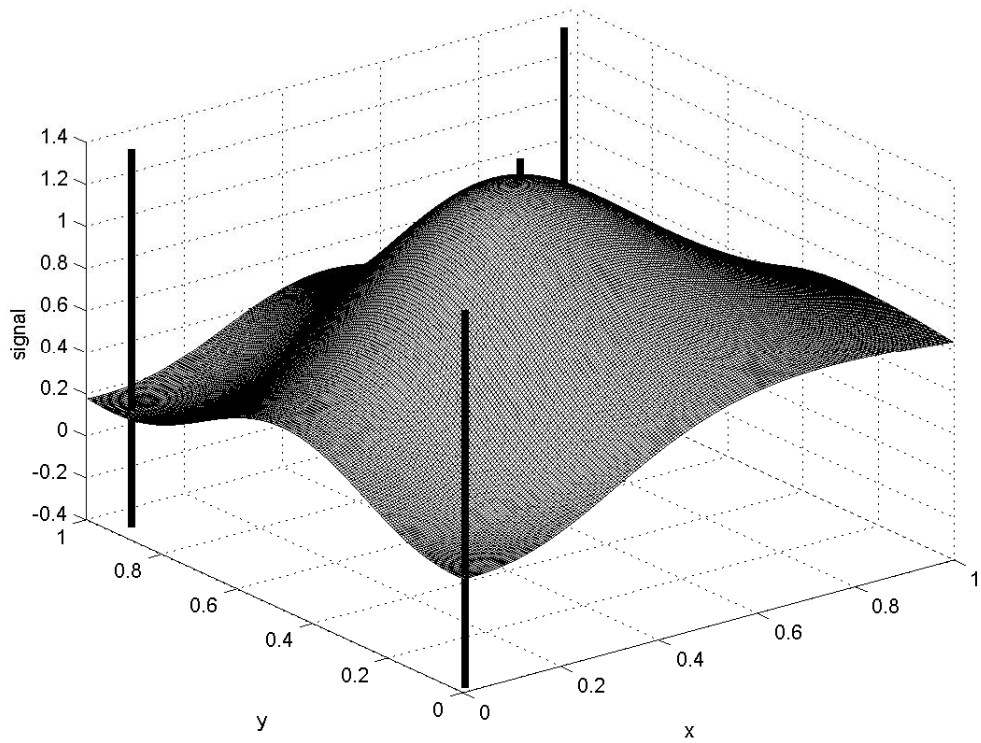


Computing Interpolation

Q: Which nodes should be selected?

A: Select the node with the largest error.

4 data points.

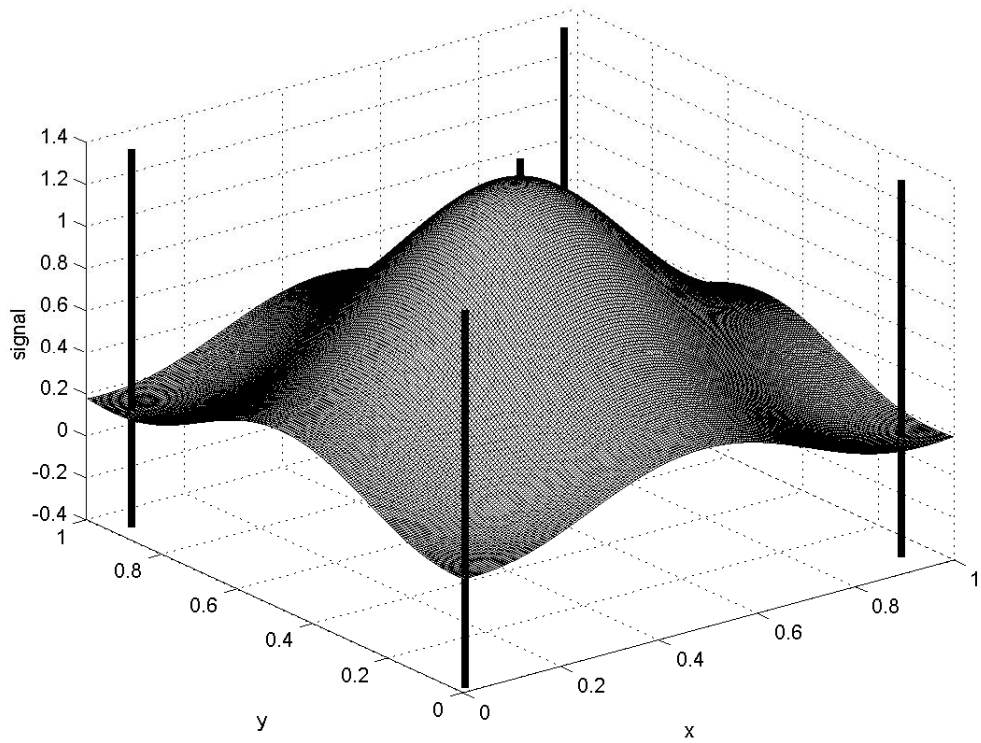


Computing Interpolation

Q: Which nodes should be selected?

A: Select the node with the largest error.

5 data points.

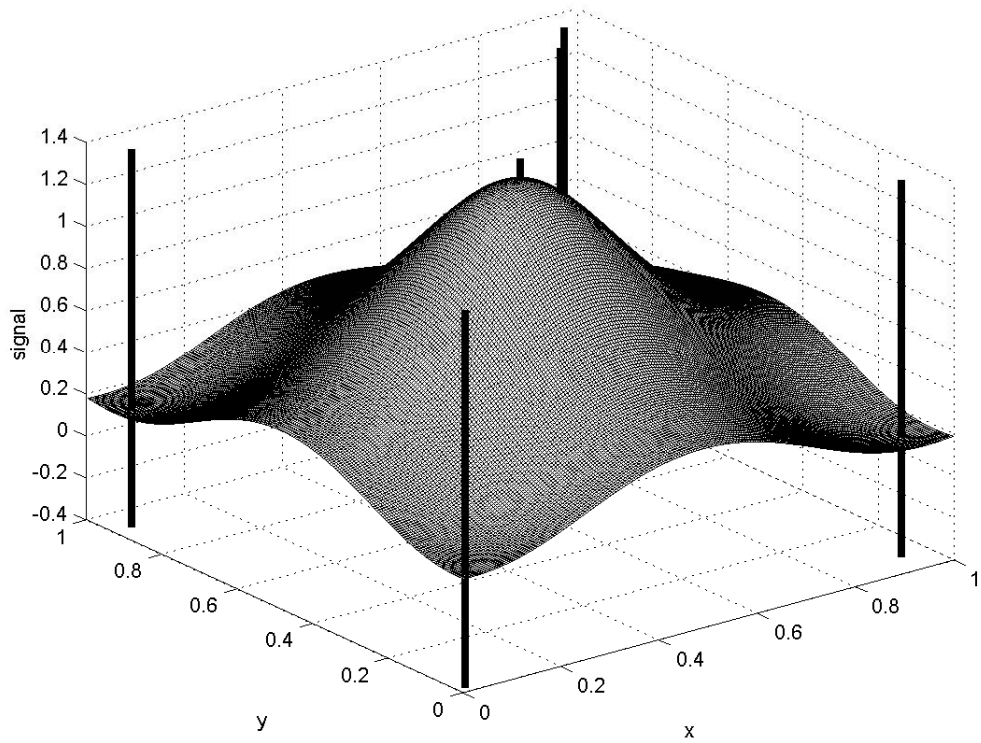


Computing Interpolation

Q: Which nodes should be selected?

A: Select the node with the largest error.

6 data points.

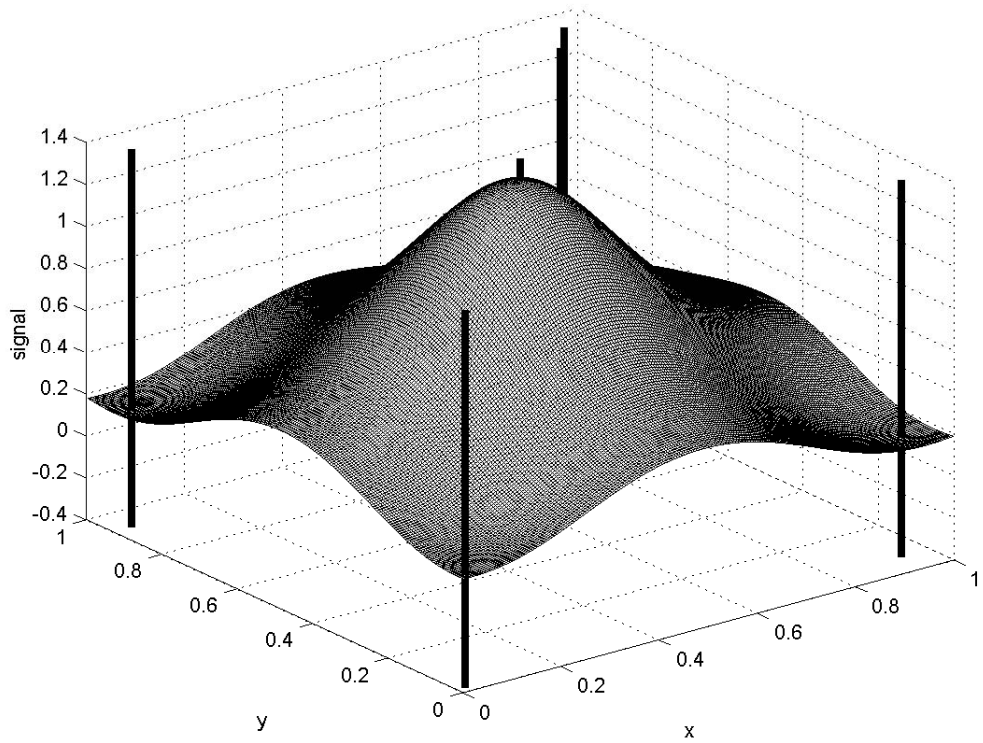


Computing Interpolation

Q: Which nodes should be selected?

A: Add a data point step by step. Use a prioritized MAC protocol and compute priority=1/error.

6 data points.



Does this really work in practice?

Conclusion

Conclusion

Sensor Readings

+

A prioritized MAC protocol

=

Cyber-Physical System with good performance

Thanks you for listening!