sensornet tasking in the large

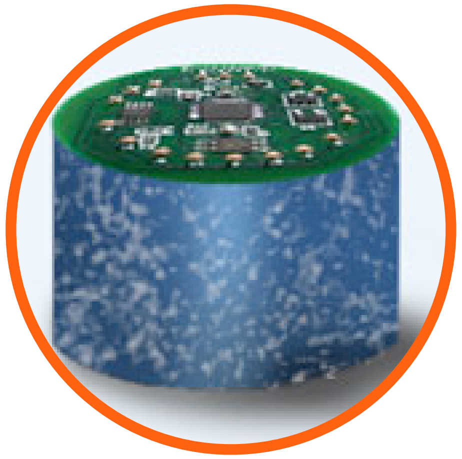
querying, inference, etc.

joe hellerstein

w/thanks to amol deshpande, carlos guestrin, wei hong, sam madden, mark paskin, wei wang

(caveats)

- this is mostly stuff we *want* to do
- there's lots of stuff we have done
  - http://telegraph.cs.berkeley.edu/tinydb



TinyDB

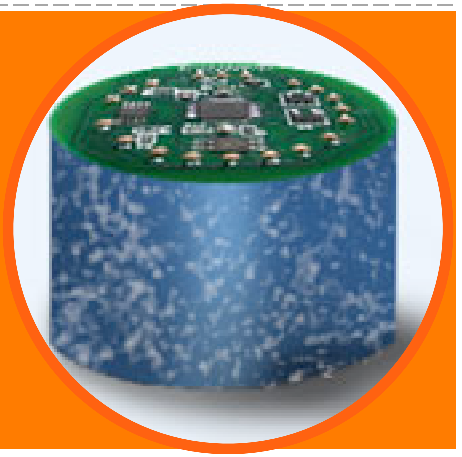
programming sensornets

- distributed and embedded programming
- data oriented
- one metaphor: real world as a database
    - declarative queries
    - automated optimization
    - query processing = routing!

querying = routing = code

- part of a bigger nets/dbs agenda
    - theme: declarative programming for large, unpredictable networks of machines
        - see also p2p work like chord, bamboo, pier, etc.
    - codd’s data independence, recast

TinyDB



- joint UCB/Intel research effort
- part of the TinyOS/NesC/TinyDB package
- continuous SQL queries over a virtual table in time
- one benefit: in-network processing
    - do aggregation at each hop of data routing
    - save BW, save power

problems with the metaphor

- discrete samples of continuous phenomena
- non-uniform sampling
- noise and loss

- raw data *requires* interpretation

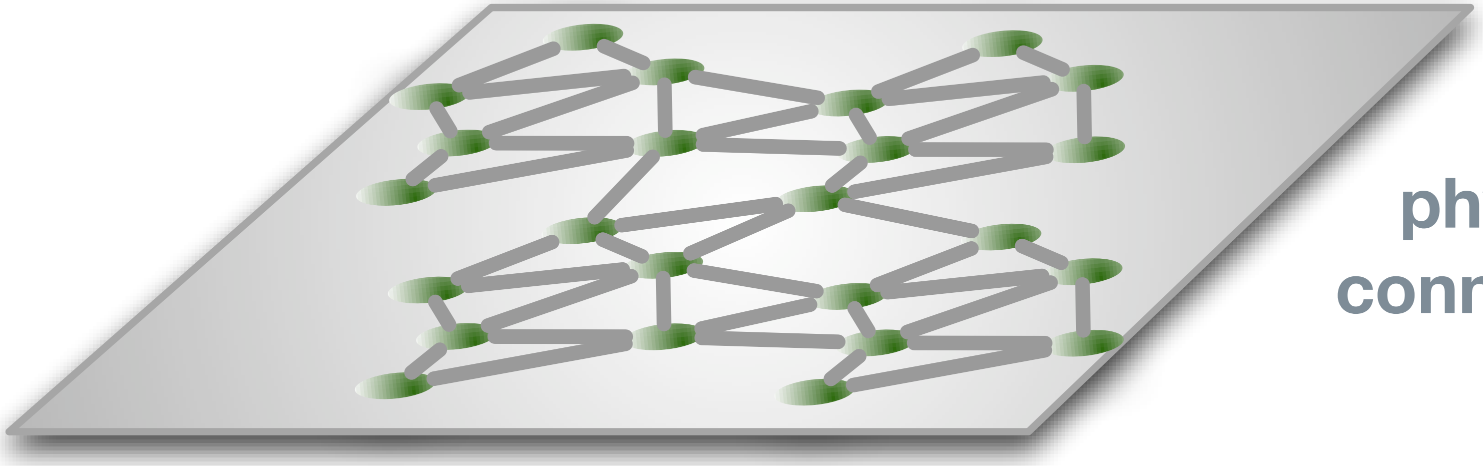
emerging agenda

- a declarative mass programming infrastructure that EMBEDS
  - models (physical and/ro statistical)
  - inference in the network
  - coding
  - online dynamics
  - all in a reusable “query optimization” framework

queries on networks

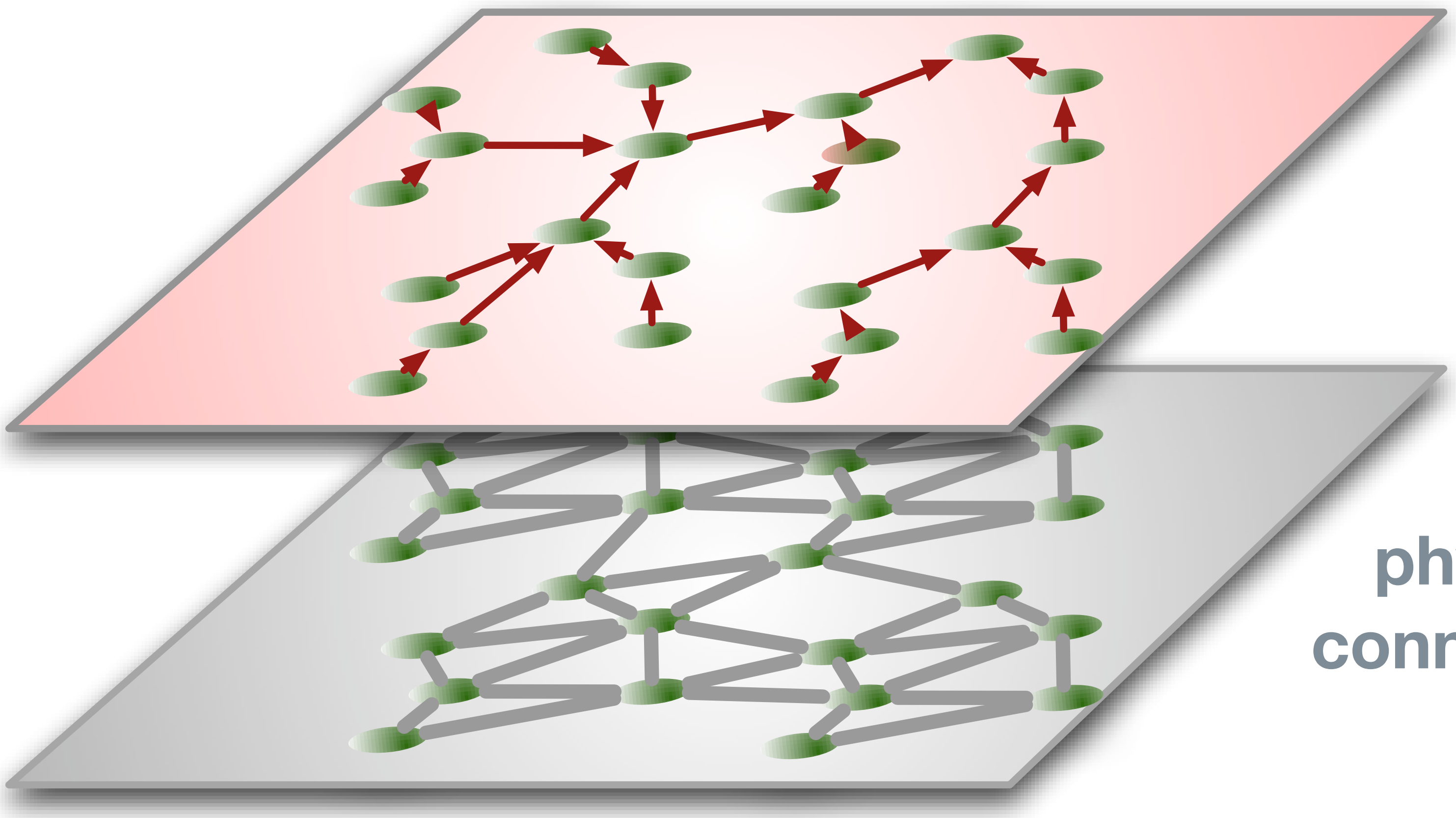
- a multi-layer optimization problem
- with:
  - error/loss tolerance
  - approximation
  - online adaptivity

optimization layers



physical connectivity

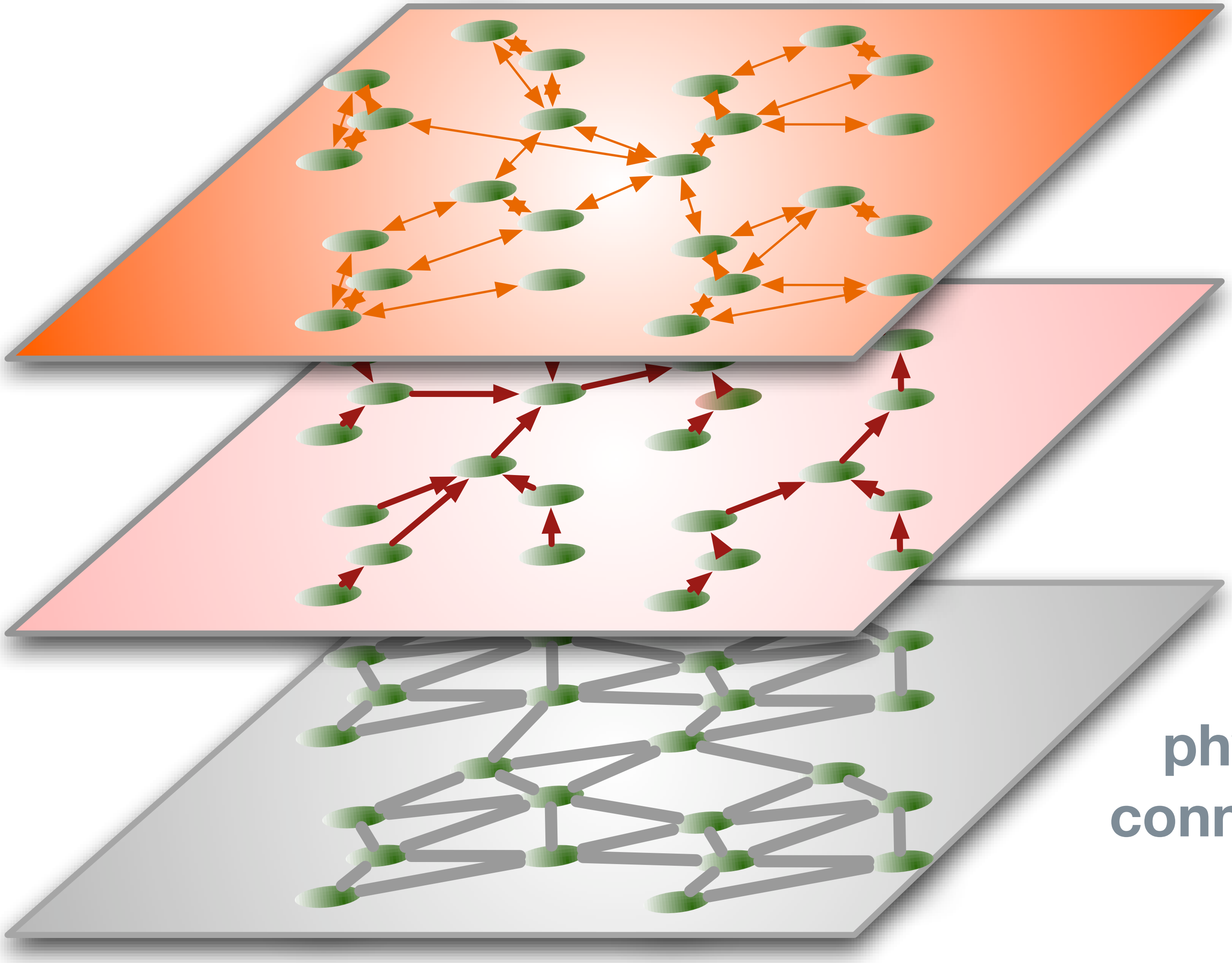
optimization layers



routing tree

physical connectivity

optimization layers

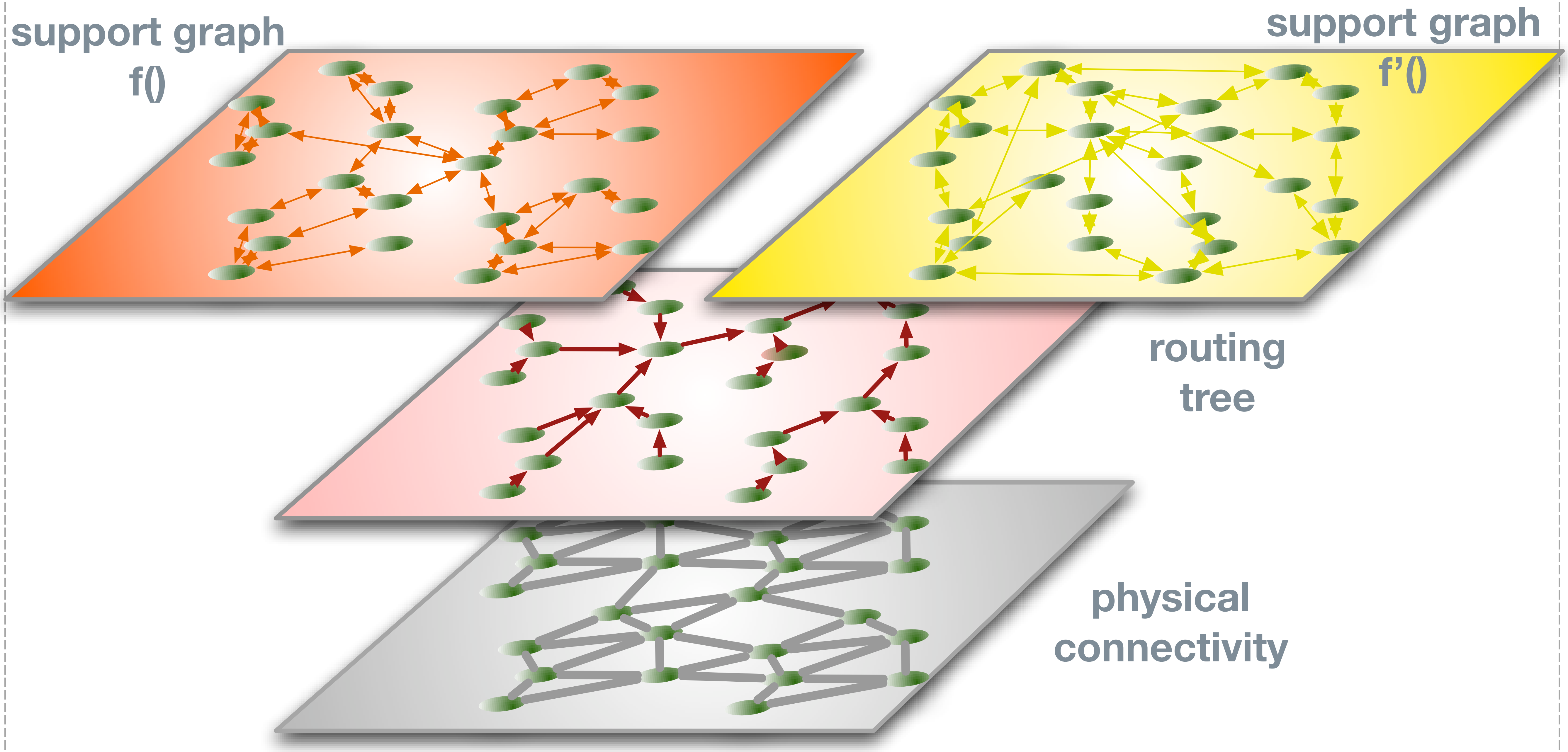


support graph
f()

routing
tree

physical
connectivity

optimization layers



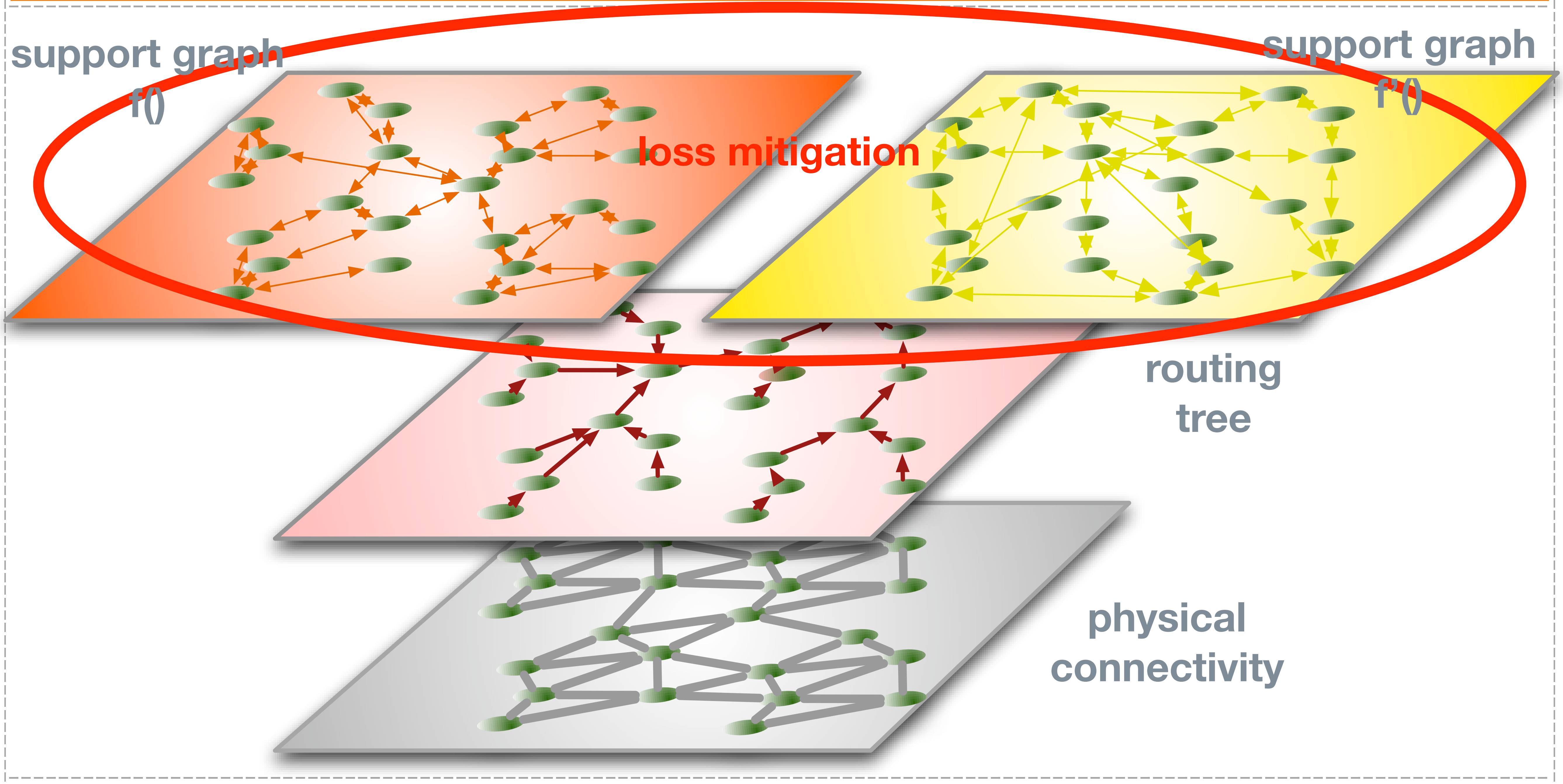
support graph f()

support graph f’()

routing tree

physical connectivity

optimization layers



support graph f()

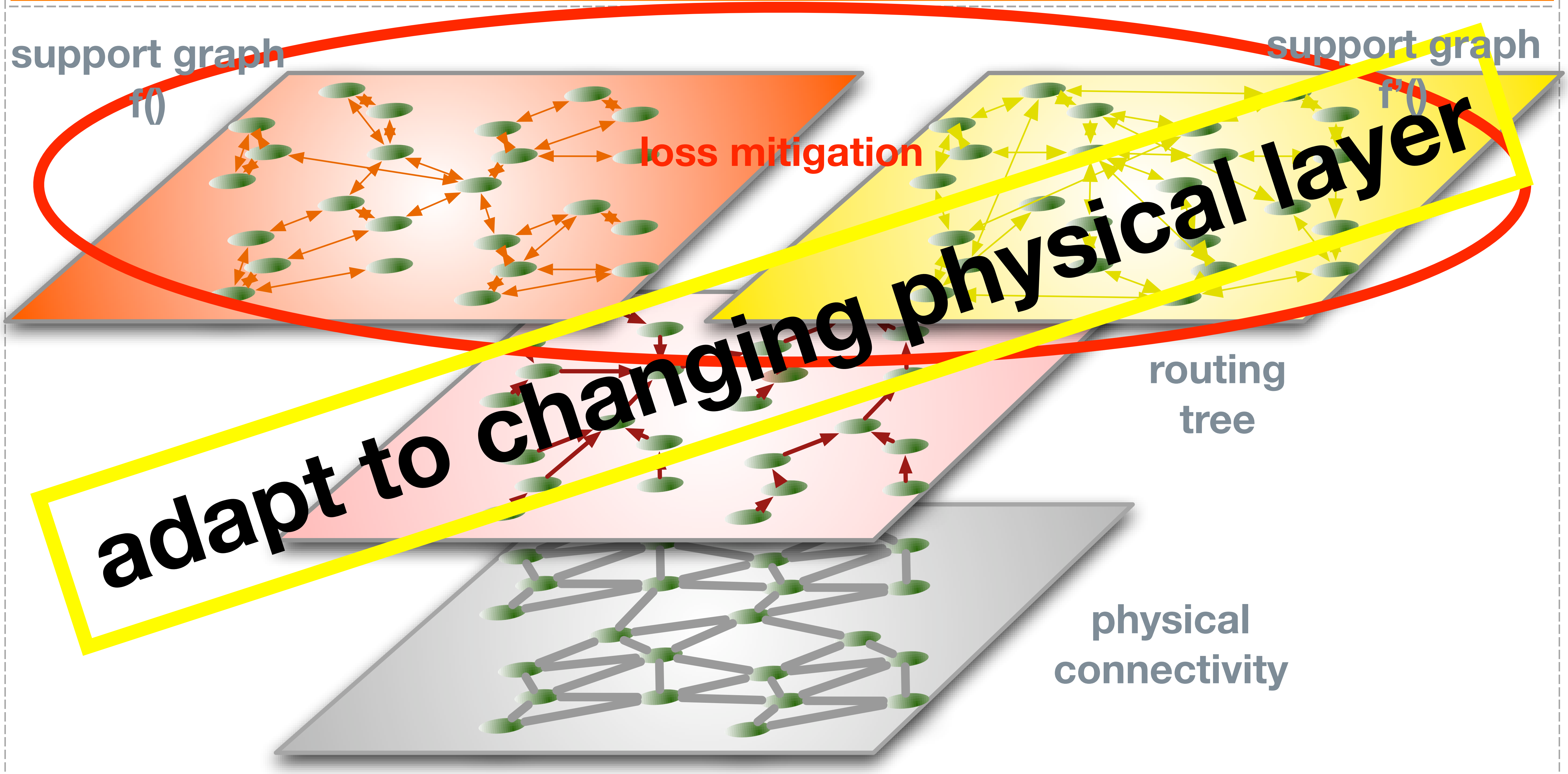
loss mitigation

support graph f'()

routing tree

physical connectivity

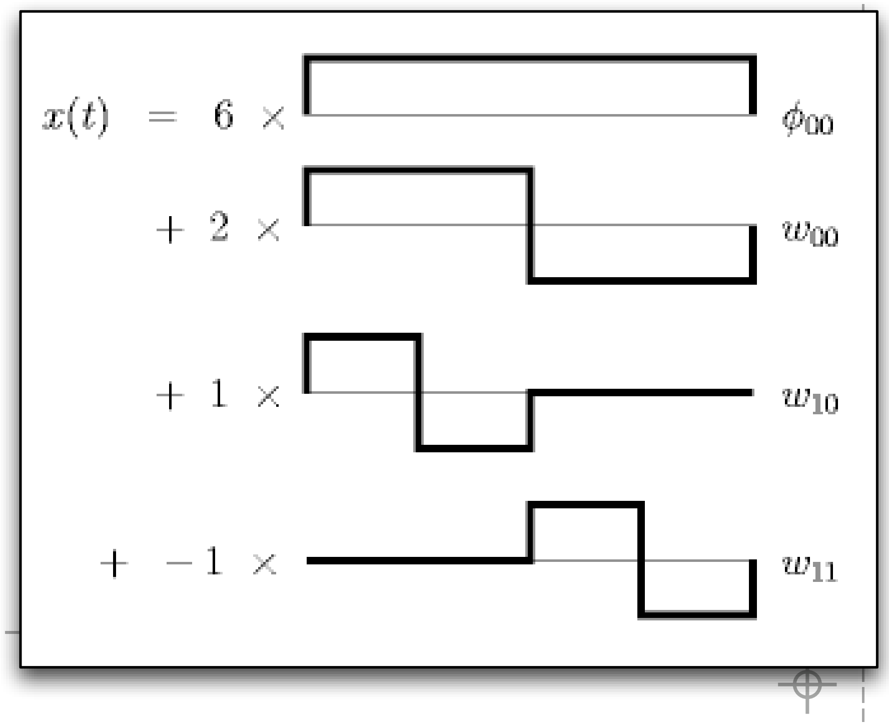
optimization layers



adapt to changing physical layer

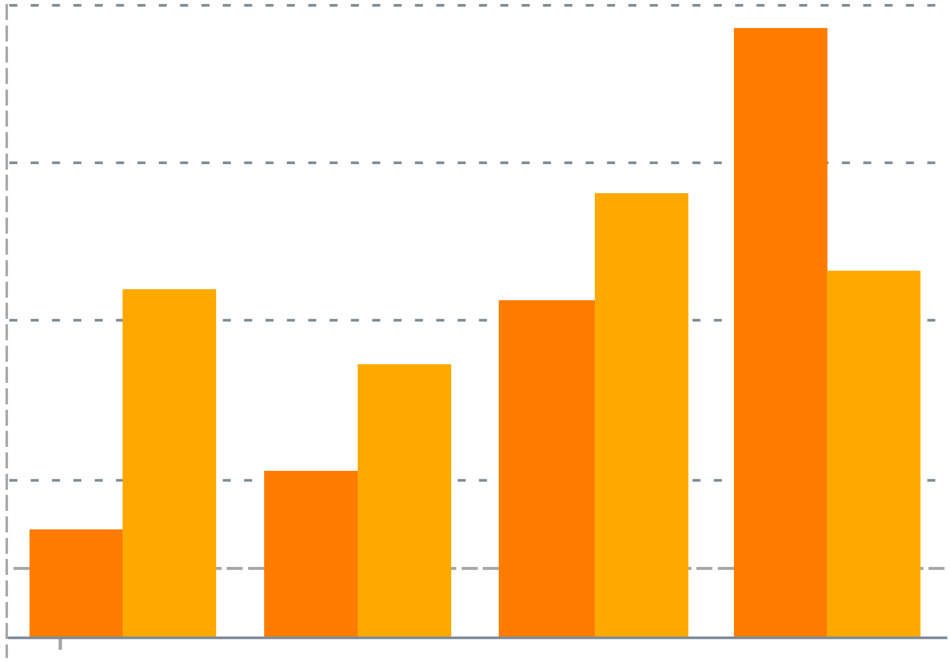
an example: wavelets

- SELECT haar(x, y, temp) FROM SENSORS
- biggest coefficients ⇒ approximate reconstruction
  - lossy compression
  - multi-resolution
- guestrin/paskin leading efforts to extend this space to junction trees
  - bayesian inference, ffts, turbo decoding, etc.
  - raises the challenge for a query/data model!

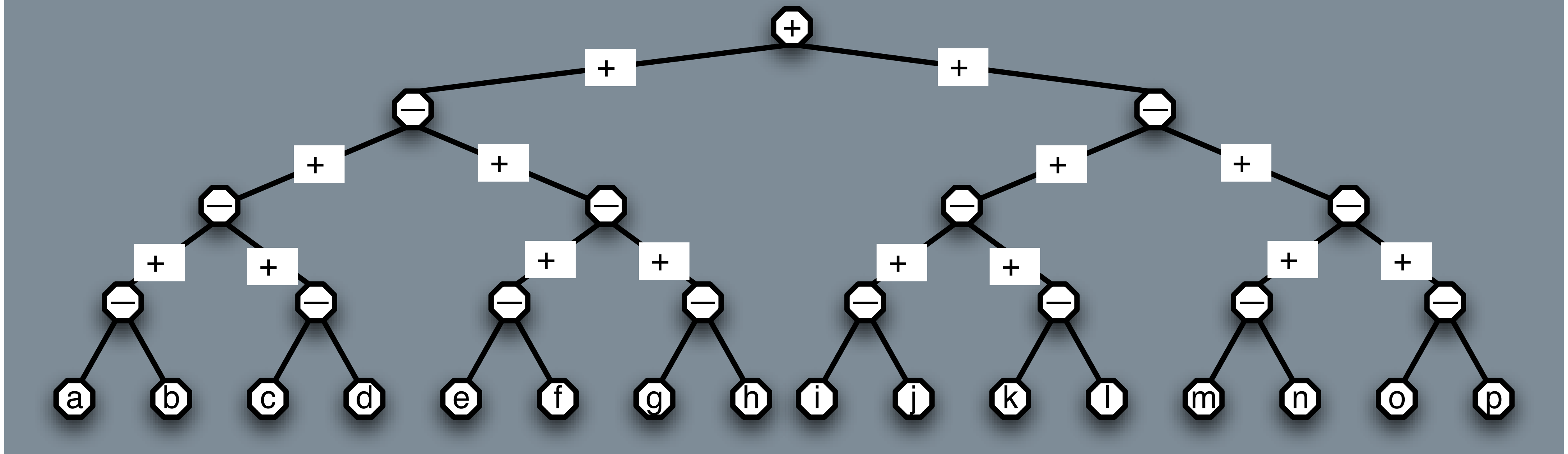


a big picture of the data

- wavelet histograms
- 2-d or 3-d (spatio-temporal) compression for reconstruction

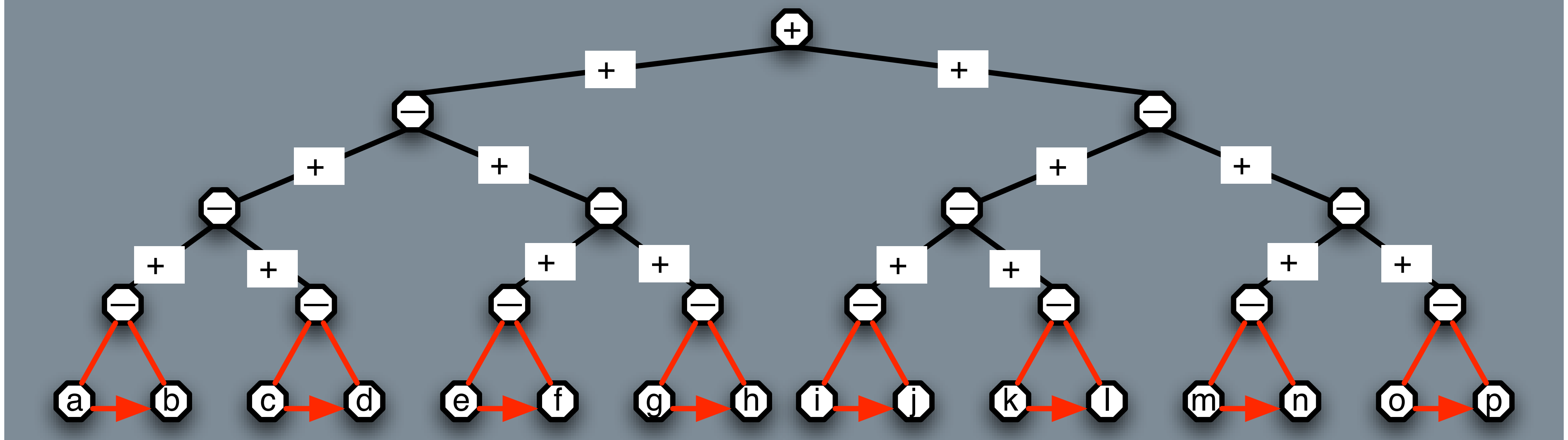


haar support graph



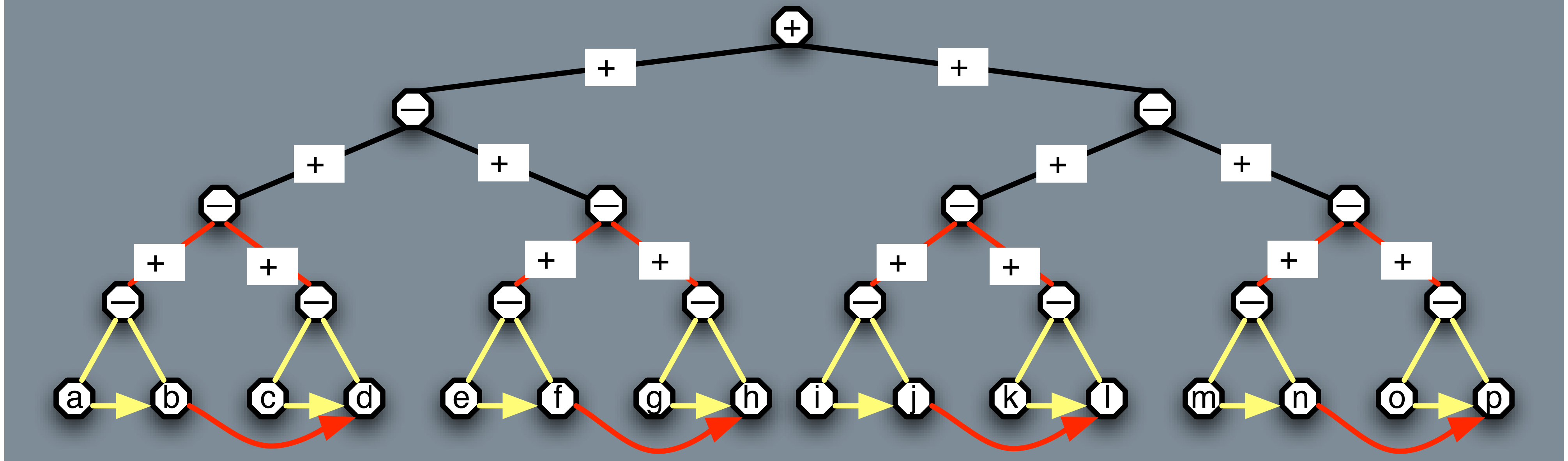
and one possible communication graph

haar support graph



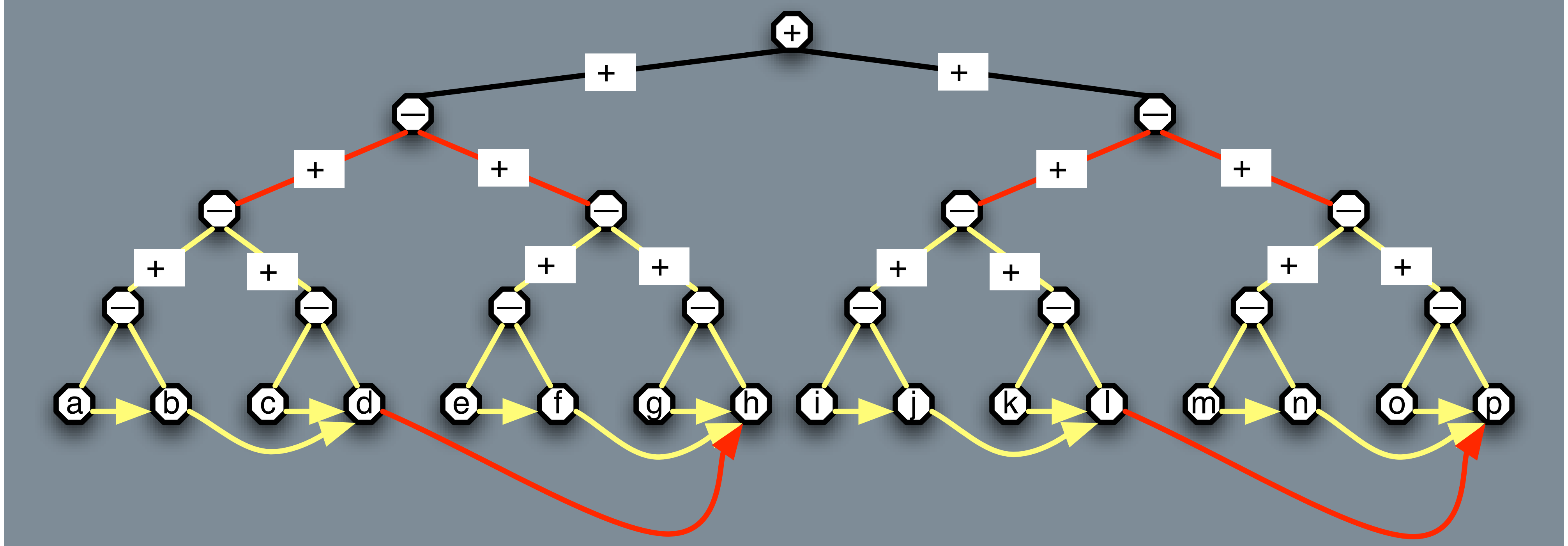
and one possible communication graph

haar support graph



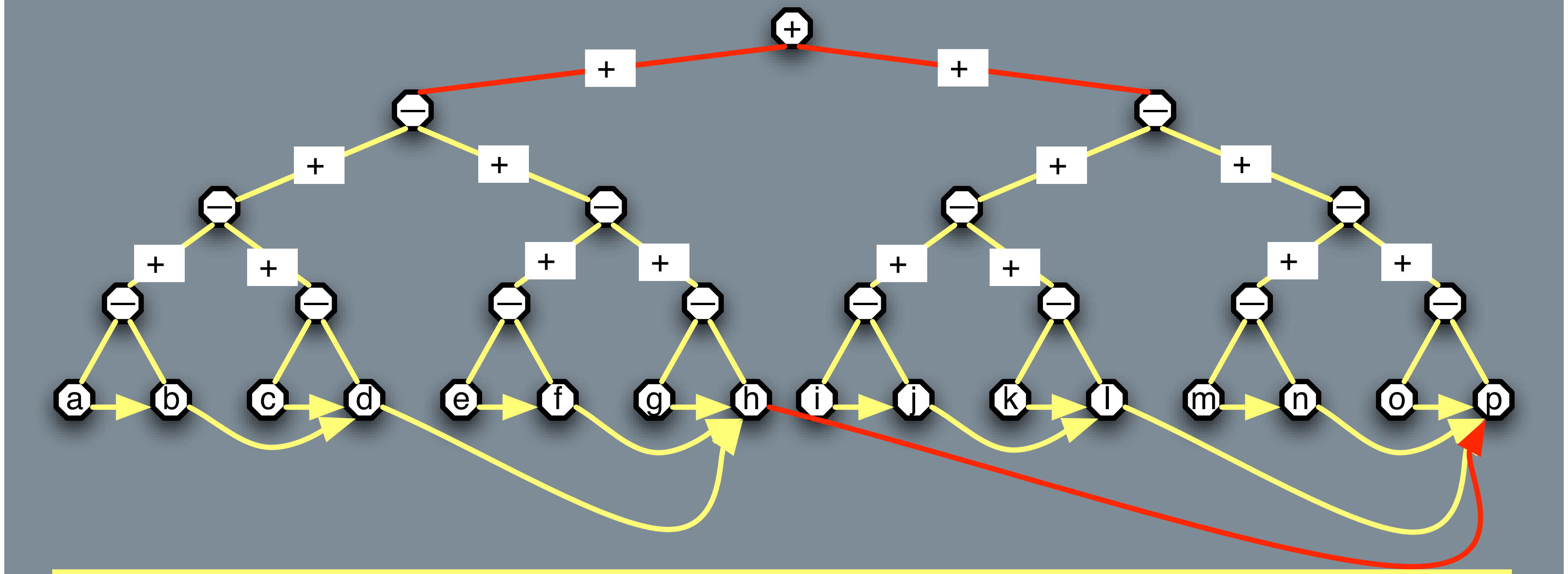
and one possible communication graph

haar support graph



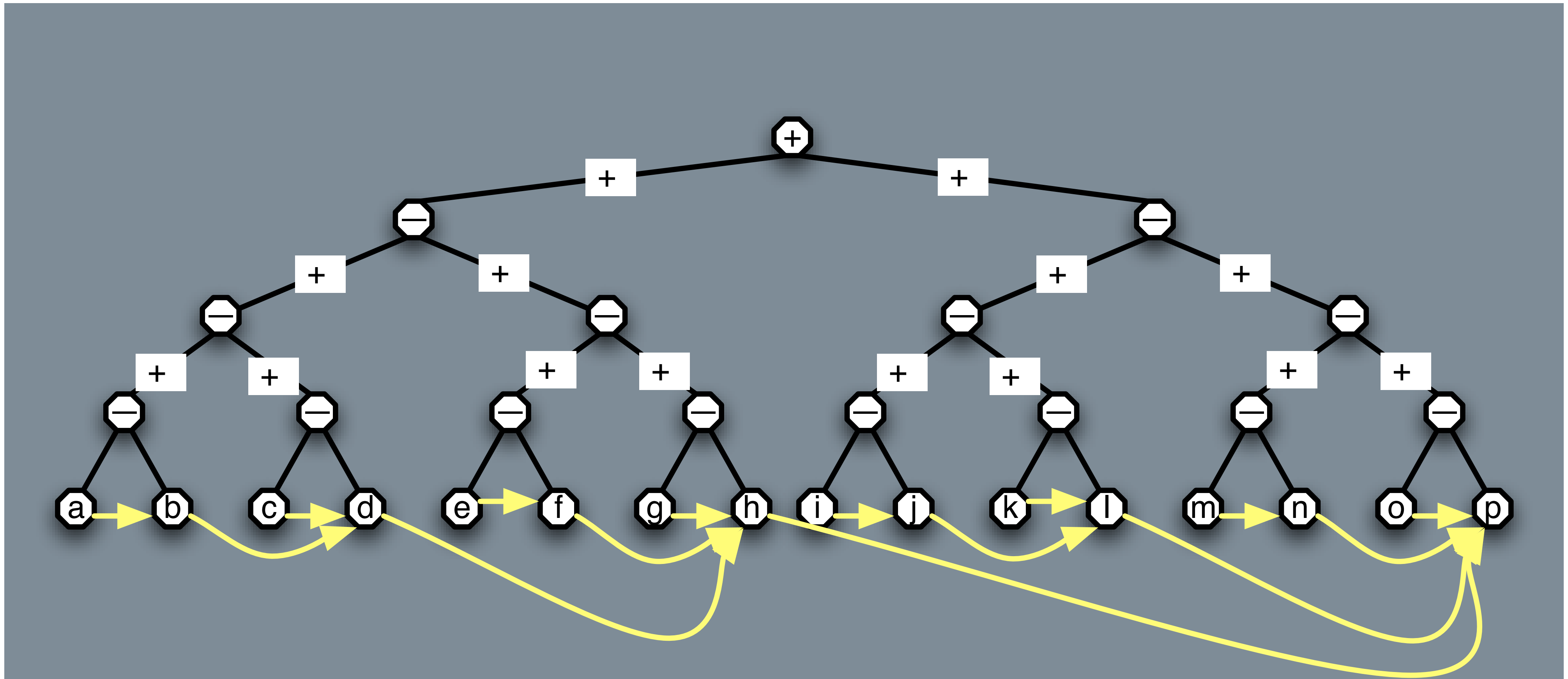
and one possible communication graph

haar support graph



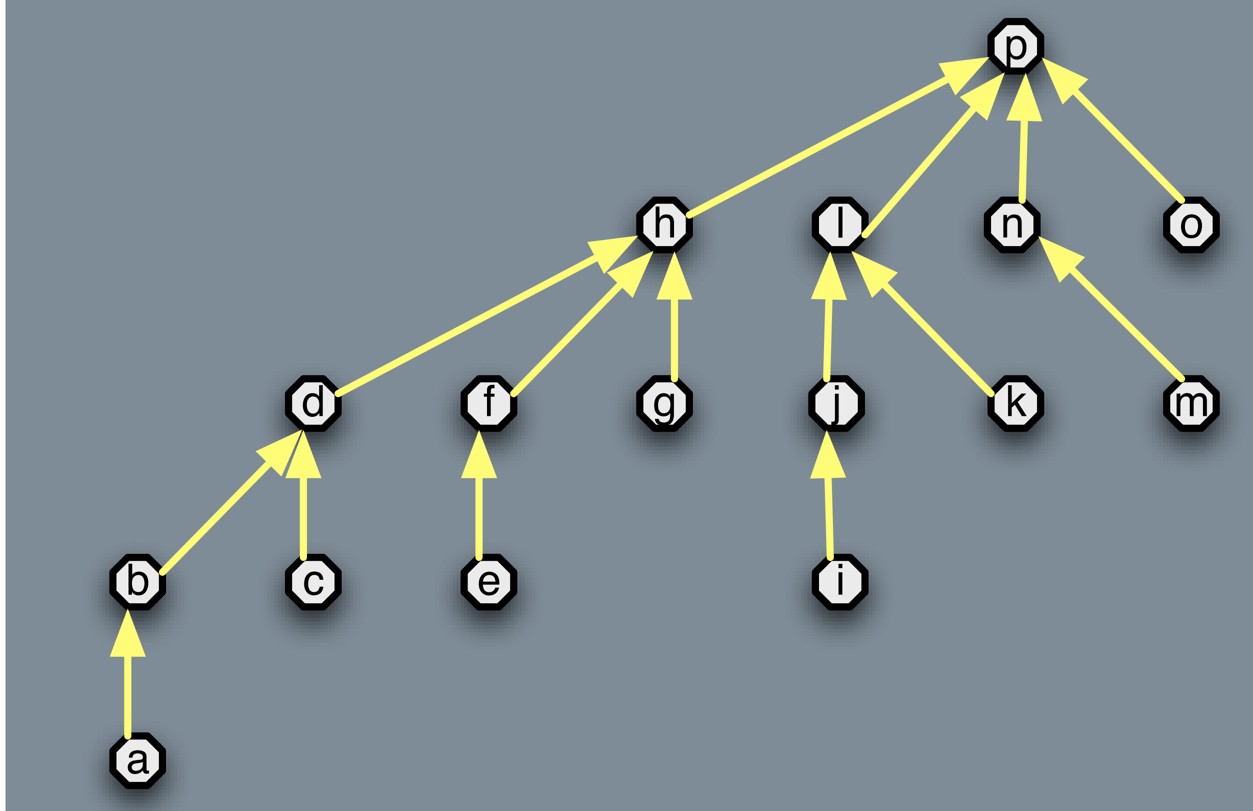
and one possible communication graph

haar support graph



and one possible communication graph

resulting comm graph



a binomial tree!

continuing the fun

- probability of a good binomial comm graph at physical layer?
- tradeoff requiring a binomial tree against coping
- tradeoff against approximate versions of haar
- loss tolerance
- online adaptivity

generalizing

- optimizing for different scenarios in a systematic way
  - remember power of a reusable declarative infrastructure!
- families of functions grouped by properties of support graphs
  - group theory as a tool here (e.g. Cayley graph routing)
- families of approximation algorithms for higher-level tasks
  - and their mappings to support graphs
- integrating across, e.g. erasure codes and approximation algs