Flooding

on recv(pkt, ifin):
    for i in interfaces:
        if i != ifin:
            send(pkt, i)

Deliver Packets ✓

But

Cycles Bad

Source: h0
Destination: h2
Packets everywhere all the time.
How to solve?
- Remember all packets forwarded
  - How many packets should be remembered?
  - How much memory?

- Change graph so there are no cycles

How to change graphs?

Remove links so that no cycles remain → Without Disconnecting Graph
Example

A -- D

h0 -- B

C -- h2

h1

h2

h1

h0

h5

h3

h4

D -- E

F
Q: How to remove links?

Assume all networks are designed to be acyclic

\[ \downarrow \]

Done by someone else.

Does not work in many environments.
Automate?

Modified Flood

deactivated = Set()

on recv(pkt, ifin):
    if ifin in deactivated:
        return
    for i in interfaces:
        if (i != ifin and i not in deactivated):
            send(pkt, i)
Algorithm that converts graph with cycles to acyclic graph?

Requirements:
1. Acyclic
2. Spanning

Root

Spanning Tree?

How?
Algorithm?
From Algorithm to Protocol

Step 1
Identifying Root

Root = id
On boot:
   Bcast((root=root)
On recv(p, i):
   If p.root < root:
      Root = p.root
      Bcast((root =
root)

A < B

Step 2
BFS
Putting it all together