Location Awareness (Route Discovery in Ad-Hoc Networks)

October 30, 2020

Outline

- Introduction
- Models
- Gabriel Test
- Geometric Routing
  - Compass Routing
  - Face Routing
- Bluetooth networks

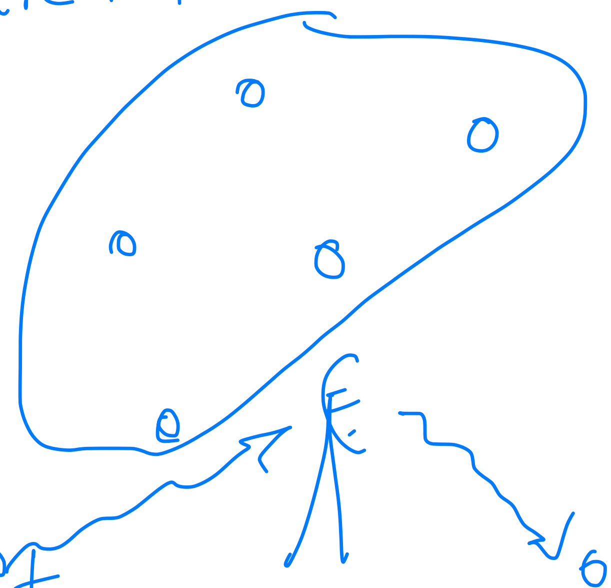
Introduction

Lots of Wireless Systems

Software Defined Radio

- Wireless systems from
  - Piconets
  - Home/Office Networks
  - (Packet) Radio Networks SDR
  - Cellular phone systems
  - Sensor Networks
  - Satellite networks Delay Tolerant

  have become all too pervasive in our everyday lives.

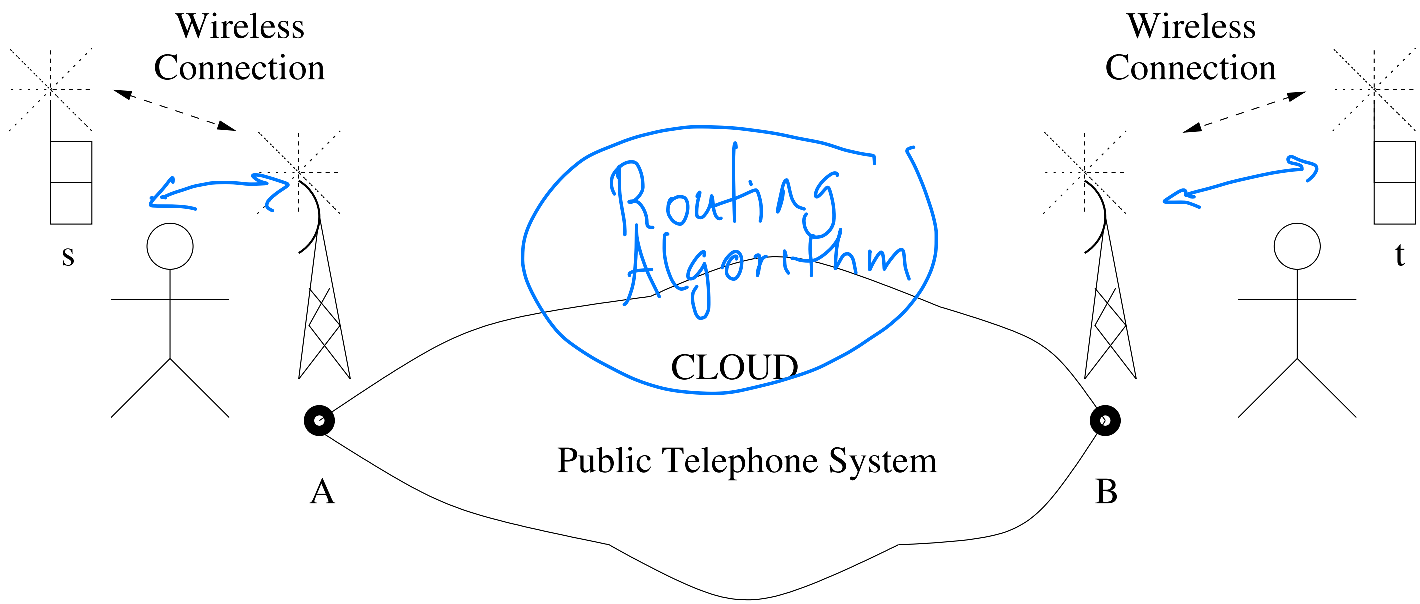


- **Questions:**
  - How do you discover a route in such a wireless system?
  - Is there a general method to find a route?

We will see later Routing Principles

The Way it is

- How do you discover a route from a source s to a destination t?

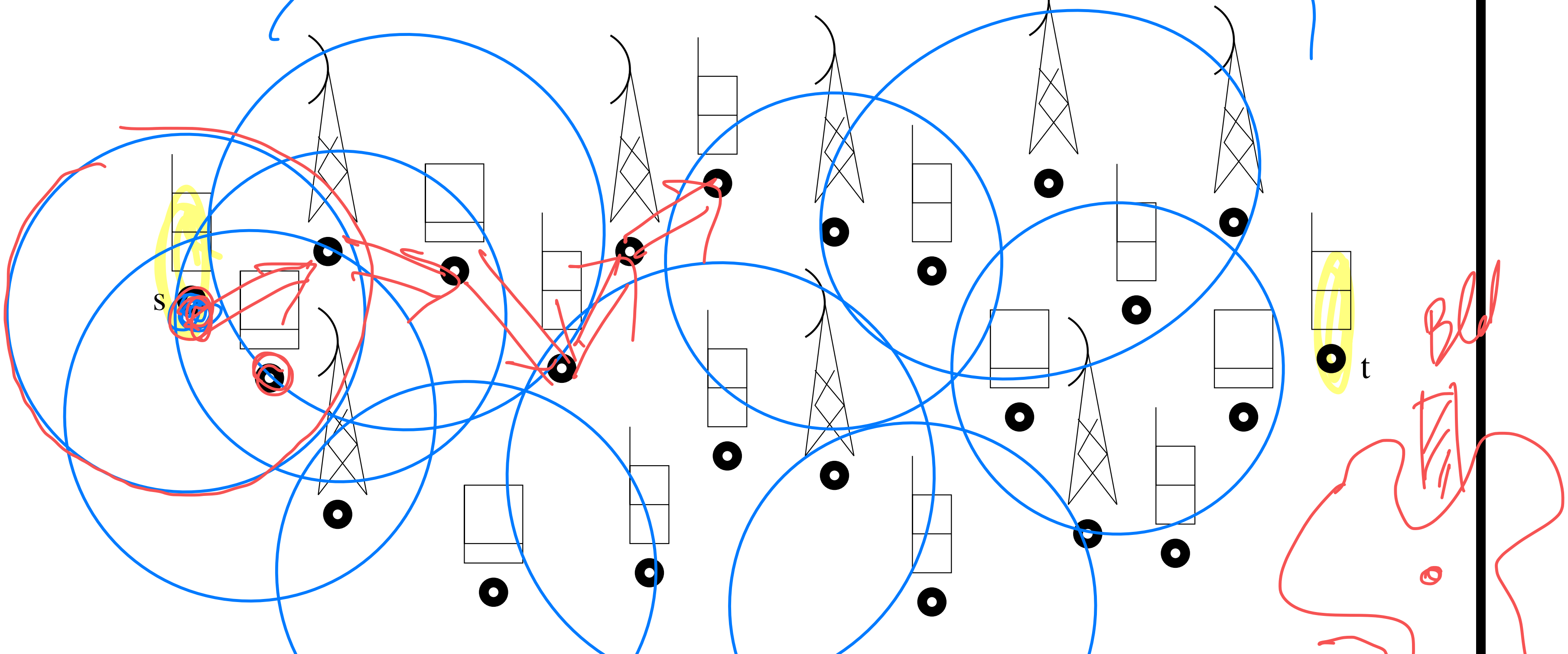


- **Simple:** 1. s sends message to base station A. 2. The public phone system transmits it to base station B. 3. Base station B transmits it to t.

Routing Data from s to t in a Wireless Ad-Hoc System

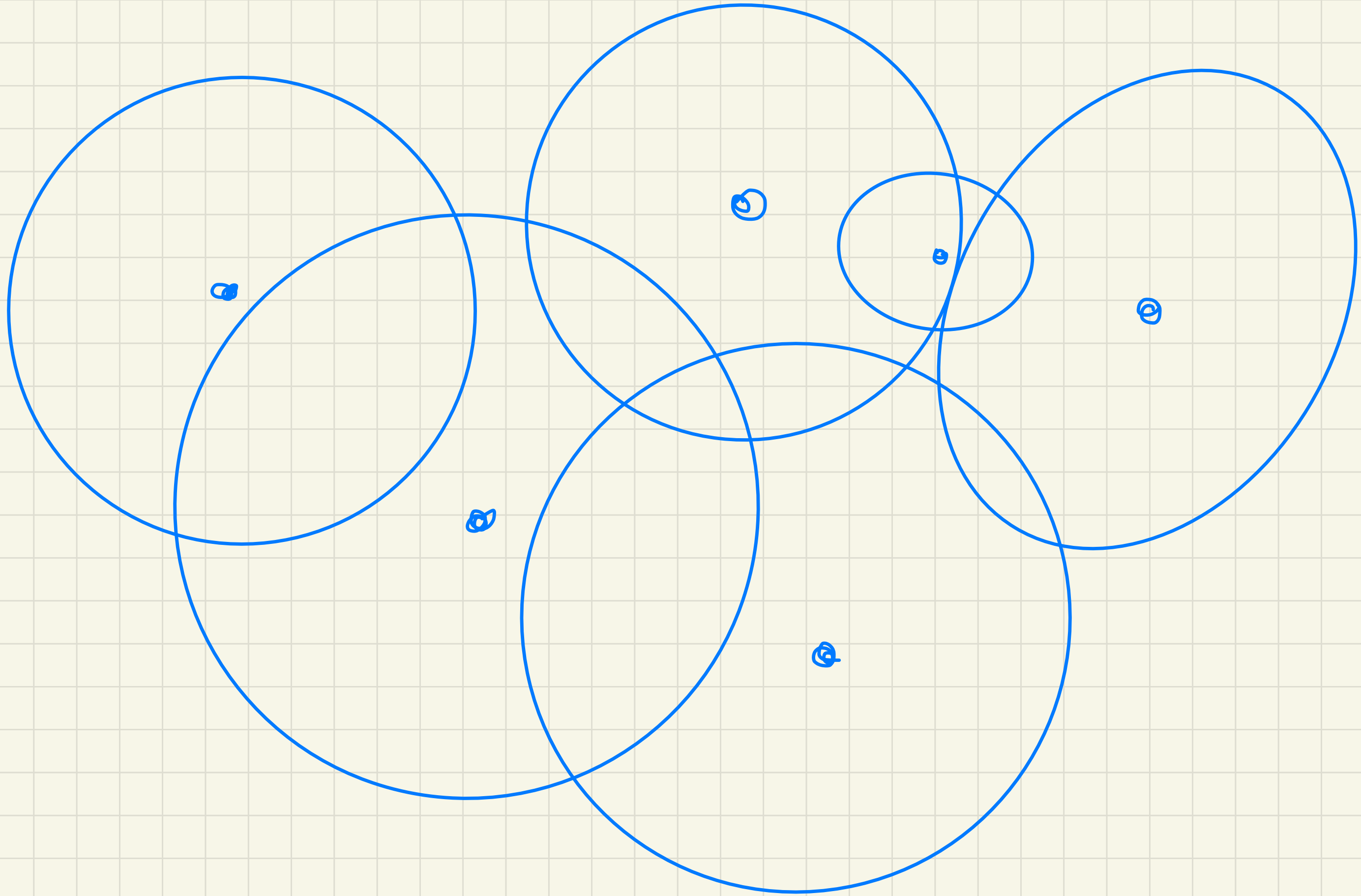
- How do you discover a route from a source s to a destination t

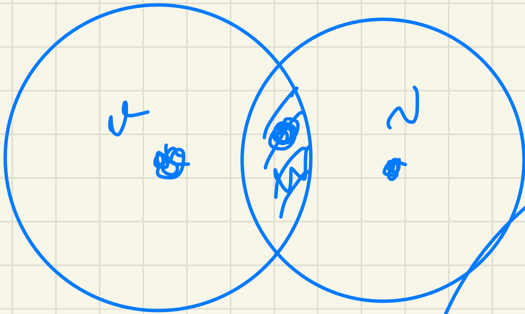
devices

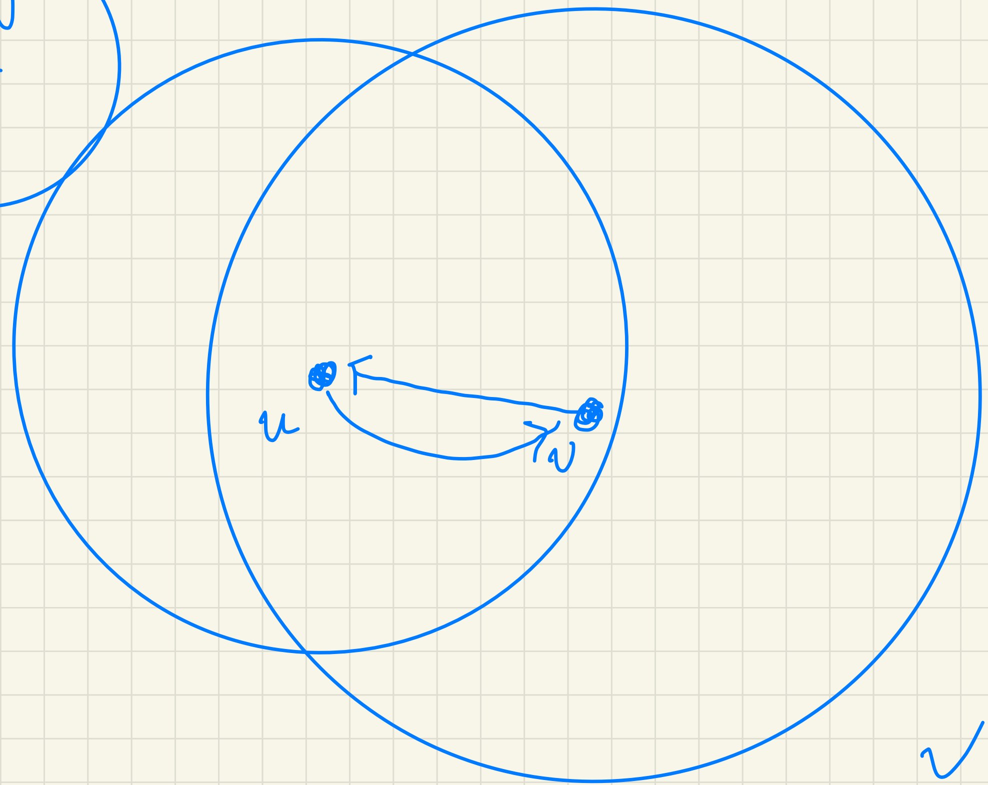


through a maze of mobile devices (radios, mobile phones, PDAs, etc) and antenas?

UE (User Entities)



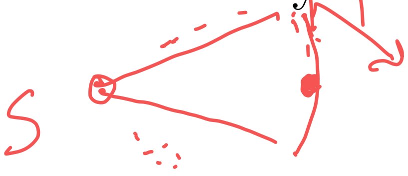

u
v

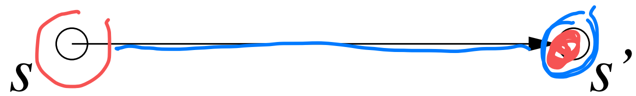

u
v
V

Realistic Models

Complexity in Models for Wireless Communication

- Traditional (wired, point-to-point) communication networks can be described satisfactorily using a graph representation.
- A station s is able to transmit a message to another station s' if and only if there is a wire connecting the two stations.

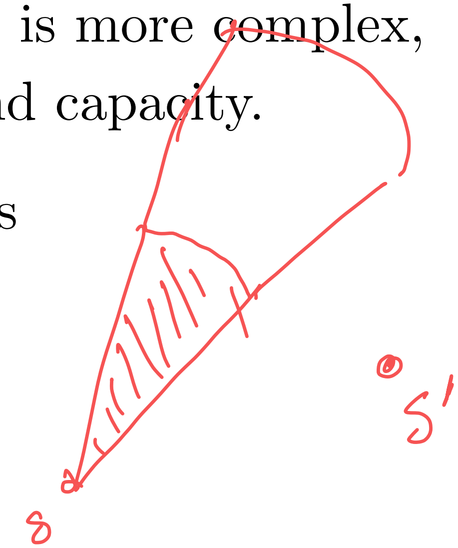




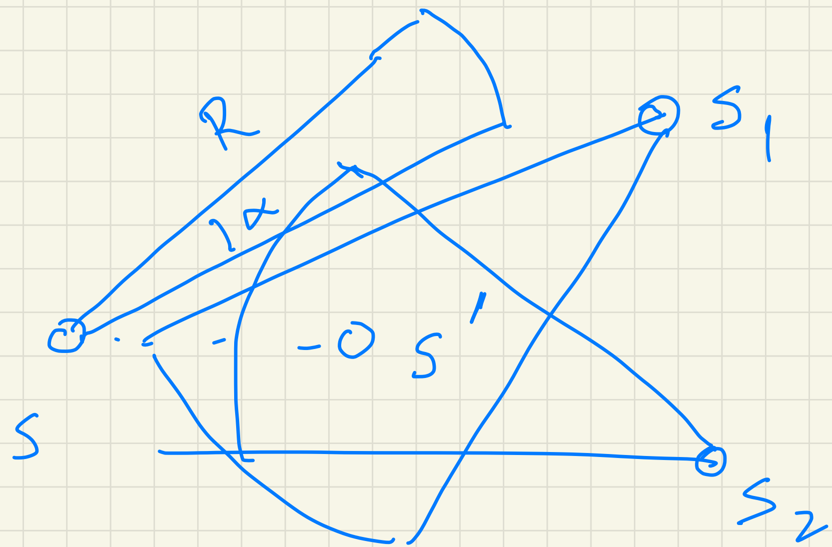
- Accurately representing a wireless network is considerably harder, since it is nontrivial to decide whether a transmission by a station s is successfully received by another station s'.
- This may depend on the positioning and activities of s and s', and on other nearby stations, whose activities might interfere with the transmission and prevent its reception

A Lot of Factors

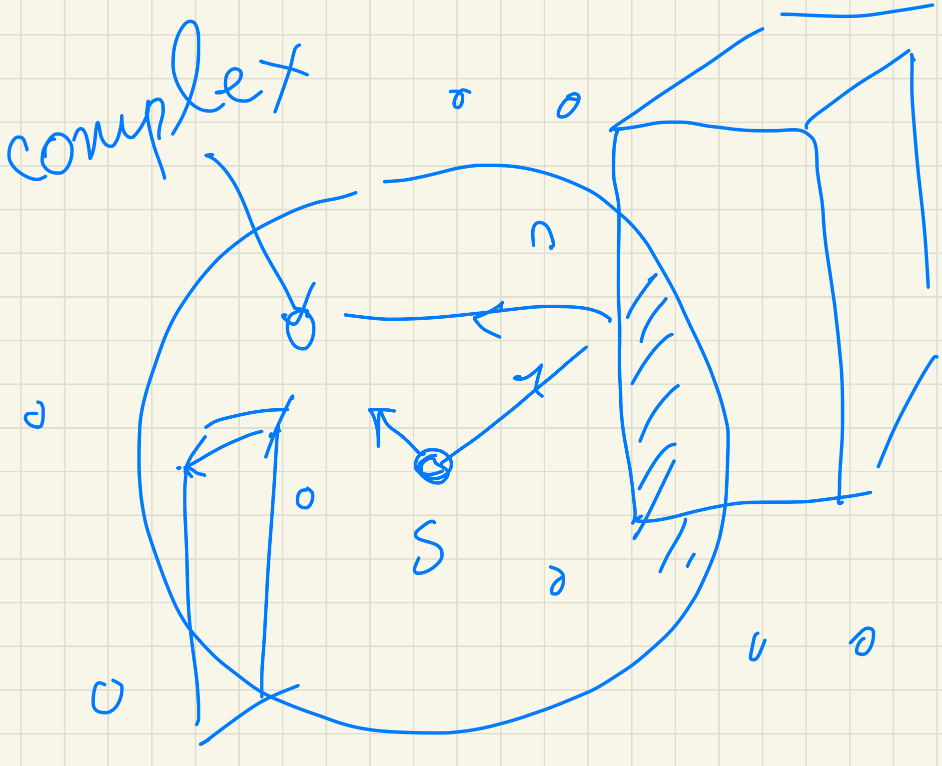
- This means that a transmission from s may reach s' in some settings but fail to reach it under other settings.
- Moreover, the question of successful reception is more complex, since connections can be of varying quality and capacity.
- There are many other relevant factors, such as
  - the presence of physical obstacles,
  - the directions of the antennae at s and s'
  - the weather, and more.



- Obtaining an accurate solution taking all of those factors into account involves solving the corresponding Maxwell equations.
- Since this is usually far too complicated, the common practice is to resort to approaches based on approximation models.



S' may not have control of who is transmitting to S'

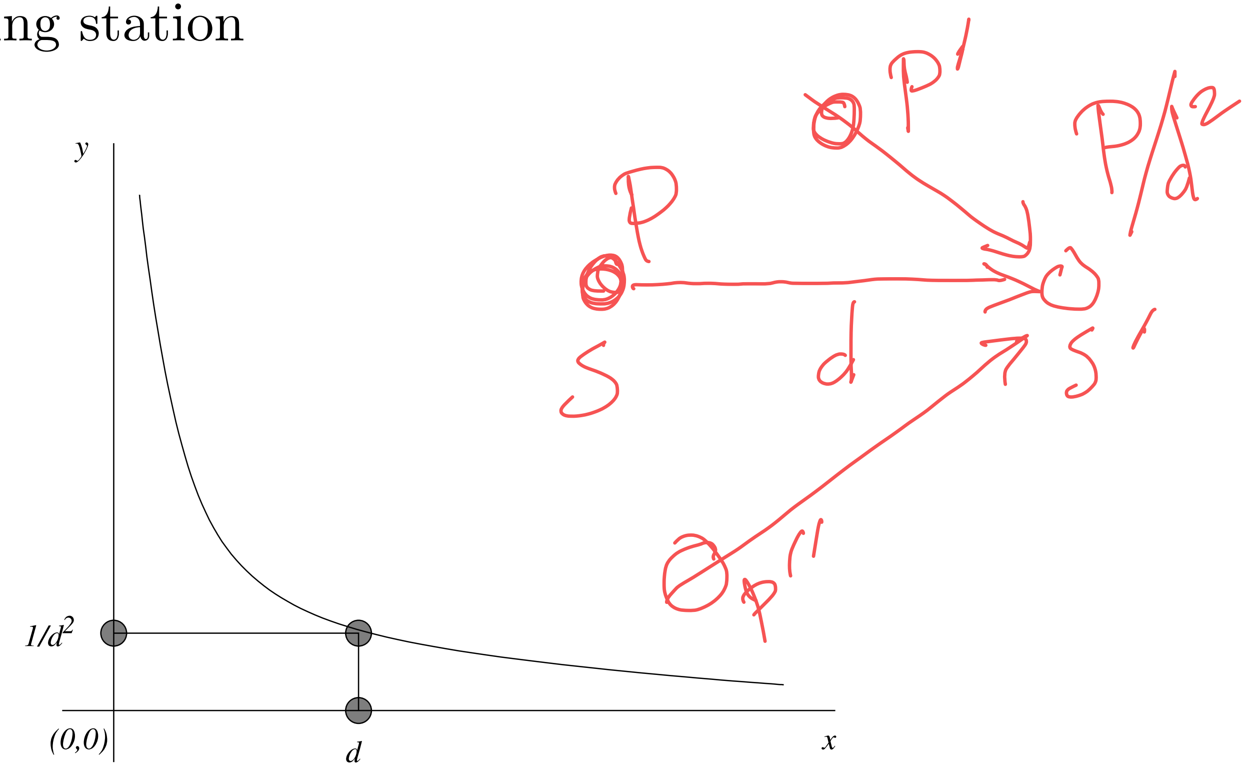


Noise: Gaussian noise

Other noise

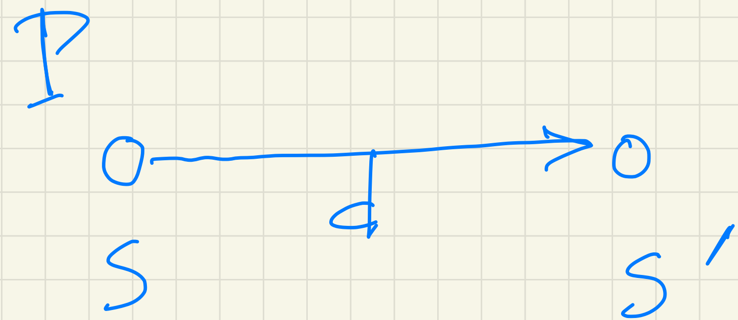
A Fact of Life: Power Assignments

- When a sensor transmits to another sensor located at distance d from the transmitting sensor, the power of the signal at the receiving station is P/d^2, where P is the power of the signal at the transmitting station



- What does it mean when $d = 0$?

s: sends a signal to s'



s': receives a signal whose strength is P/d^2

$d = \text{distance}(s, s')$

P/d^a

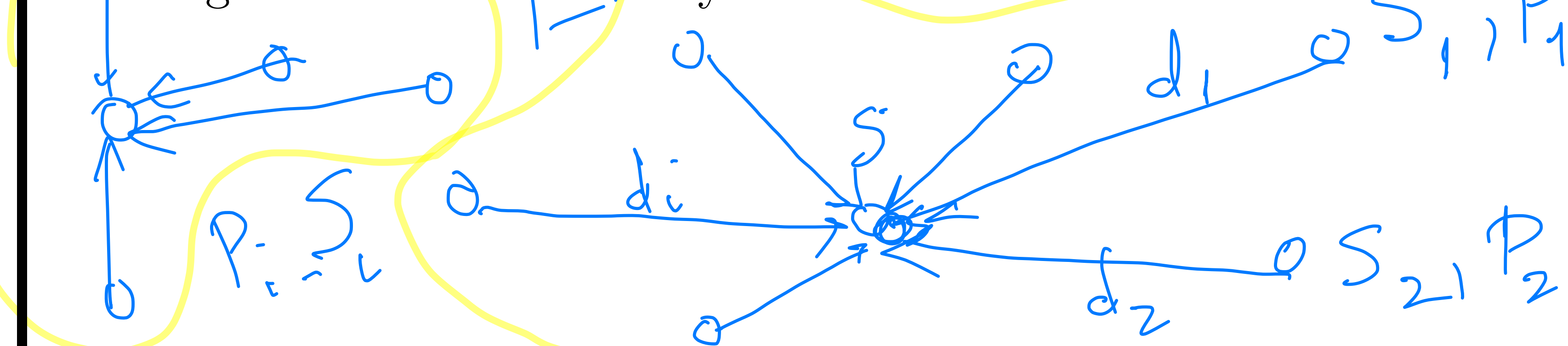
$a \geq 2$

Rayleigh's Principle: Physical Model

- Consider the setting whereby sensors $S_1, S_2, \ldots, S_k$ and S are located in the plane and suppose that sensor S_i is at distance d_i from S.

- When the signal from a transmitting station S_i reaches the sensor S it will have power P_i/d_i^2, where P_i is the power of the transmitted signal at S_i.

- When the k sensors are transmitting at the time then according to Rayleigh's principle only the most "powerful" signal can be received by sensor S.



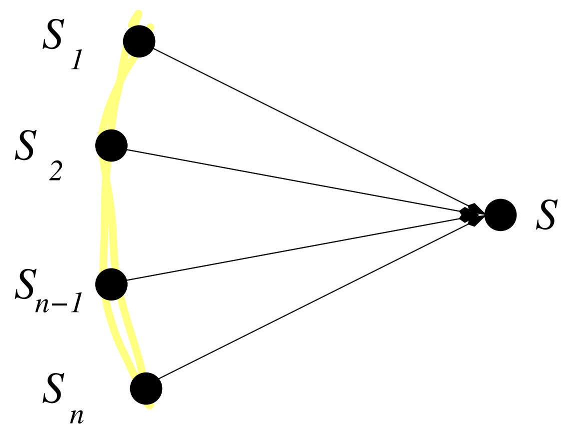
Rayleigh's Principle

- The signal from a sensor S_i, for some i, will be received by sensor S if and only if there is a threshold $\lambda > 0$ s.t.

$$\frac{P_i}{d_i^2} > \lambda \left(N + \sum_{j=1, j \neq i}^{n} \frac{P_j}{d_j^2} \right),$$

ambient noise N

which depends on technical considerations, like, sensor equipment sensitivity, and N is ambience noise.

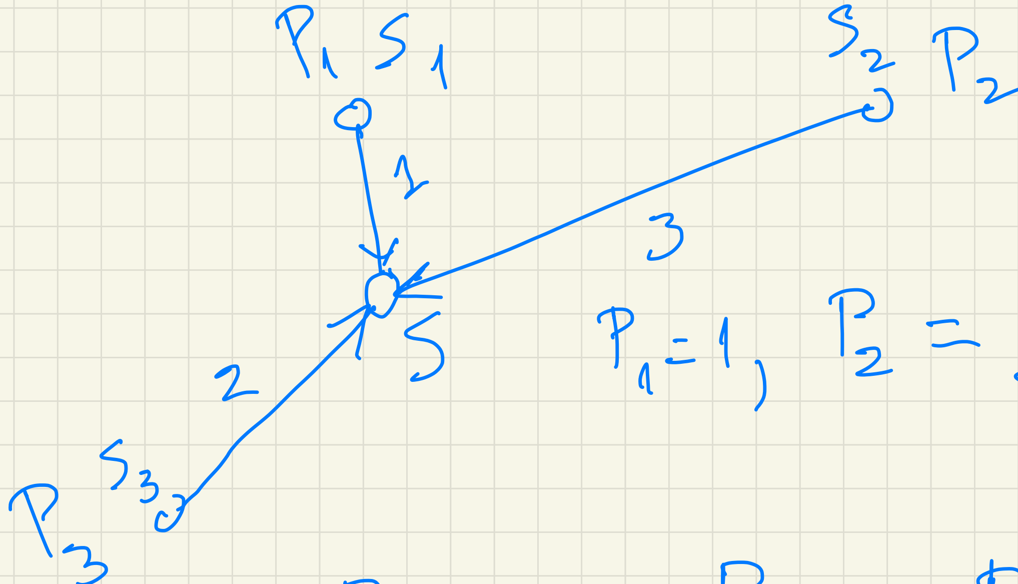


- Usually, to simplify notation, we assume that $\lambda = 1, N = 0$.

Example

$\lambda = 1$

$M = 0$



$P_1 = 1, \quad P_2 = 5, \quad P_3 = 6$

$$\frac{P_2}{3^2} > \frac{P_3}{2^2} + \frac{P_1}{1^2}$$

$$\frac{P_3}{2^2} > \frac{P_2}{3^2} + \frac{P_1}{1^2}$$

$$\frac{5}{9} > \frac{6}{4} + \frac{1}{1}$$

$$\frac{6}{4} \overset{?}{>} \frac{5}{9} + 1$$

$P_1 = 1$, $P_2 = 20$, $P_3 = 4$

$$\frac{20}{9} > \frac{4}{4} + 1$$

S_3 S_2 S_1

SINR (Signal-to-Interference & Noise Ratio)

- This formula represents a rather general model concerning the allowed transmission power, referred to as the power control model, in which each station can control the power with which it transmits.

- A simpler (and weaker) model is the uniform wireless network model, which assumes that all transmissions use the same transmission power, i.e., $P_i = 1$ for every i.

If all UEs are identical, say $P_i = 1$, $\forall i$

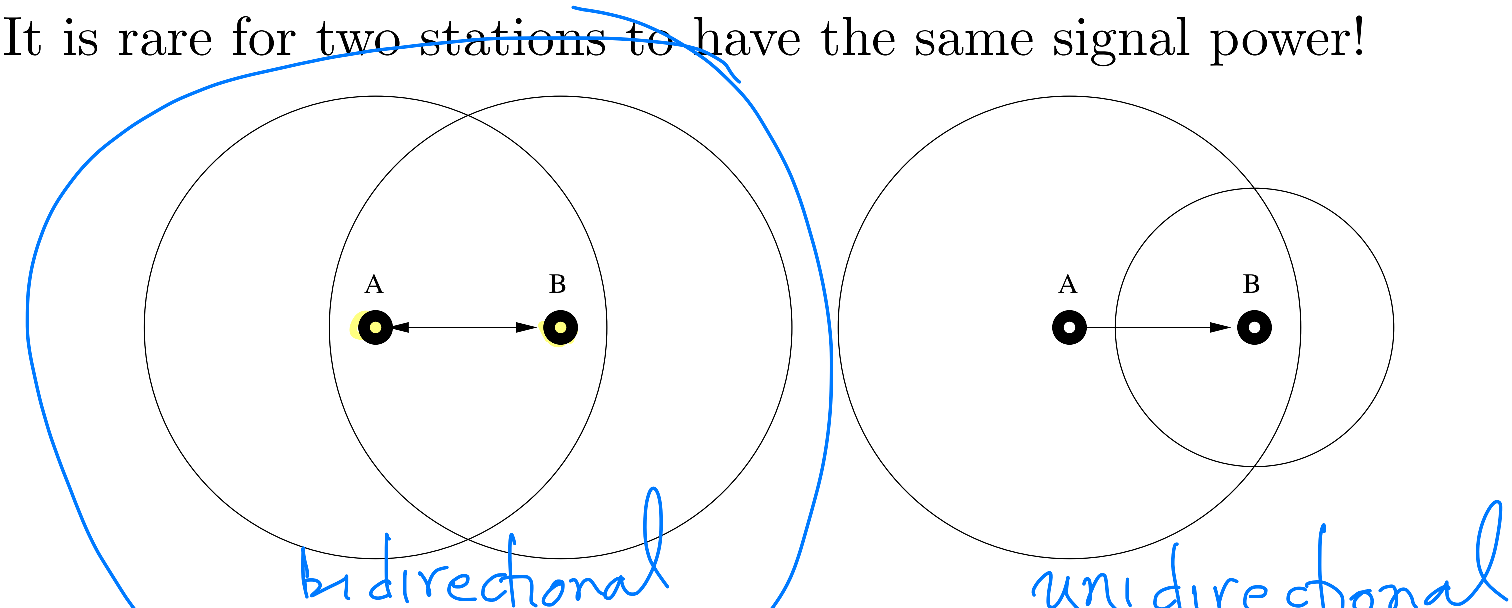
Rayleigh's Formula

$$\frac{1}{d_i^2} > \sum_{k \neq i} \frac{1}{d_k^2}$$

Idealized Models

Protocol Model: Equal Power Assumption!

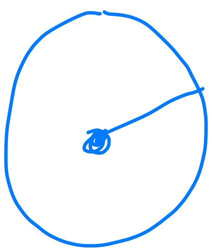
- The disk indicates an omnidirectional antenna!
- It is rare for two stations to have the same signal power!

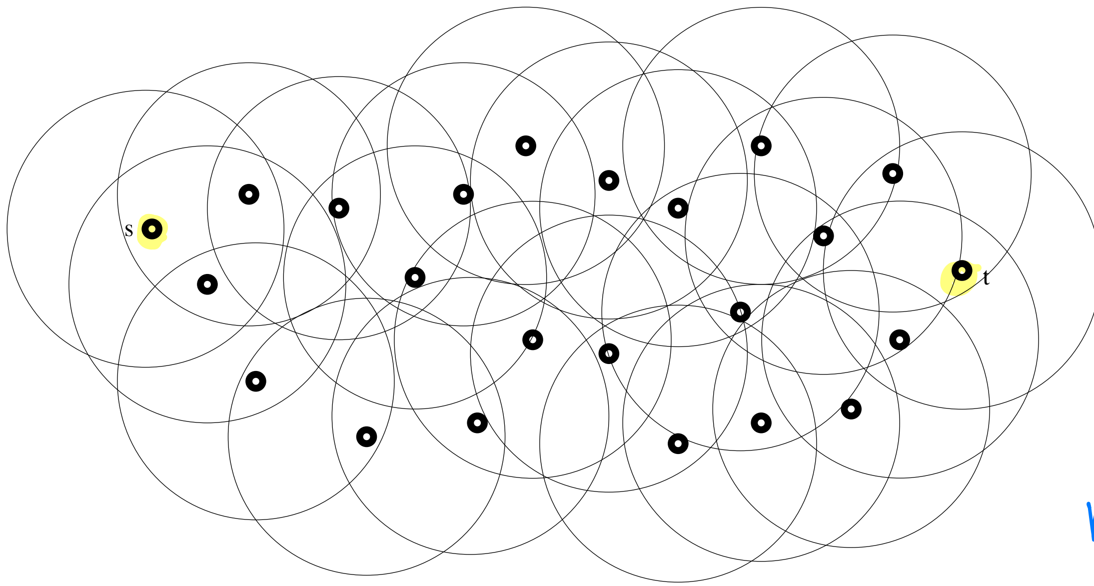


- To simplify things stations are assumed to have equal power!
- In the left picture A can reach B and B can reach A.
- In the right picture A can reach B but B cannot reach A.

From Mobile Devices to Circles

- Assuming the equal power assumption for the signals...





range=radius

...a group of circles is formed that determines network connectivity, i.e. who can reach whom!

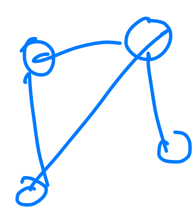
- Note that the circles have equal radius, say r: two hosts can communicate with each other if and only if their distance is at most r.

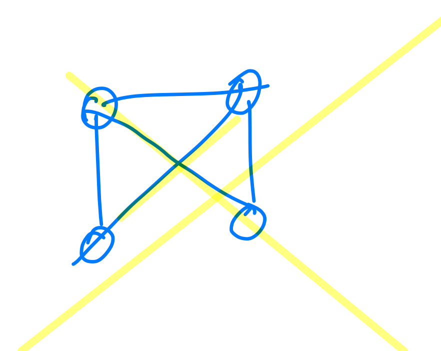
For the sake of discovering routes...

We will show...

1. ...how from a network of circles we can produce a simplified network in which edges have no crossings, and
2. how to discover routes in networks with non-crossing edges.

planar graph

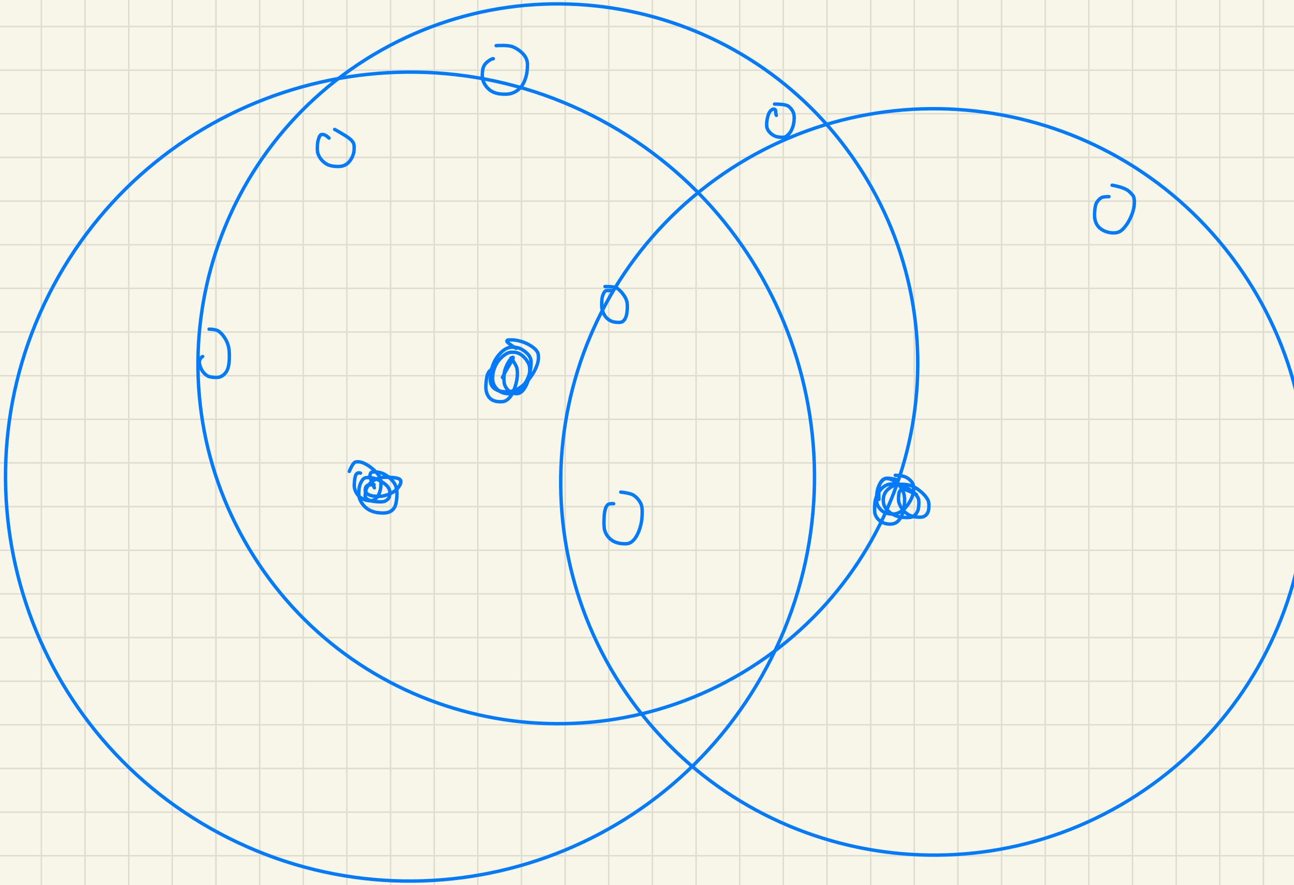




protocol

UDGs and Wireless

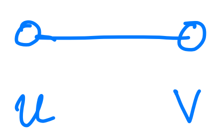
- Unit Disk Graphs (UDGs) are used in computer science to model the topology of ad hoc wireless communication networks.
- Nodes are connected through a direct wireless connection without a base station. It is assumed that all nodes are homogeneous and equipped with omnidirectional antennas.
- Node locations are modeled as Euclidean points, and the area within which a signal from one node can be received by another node is modelled as a circle.
- If all nodes have equal transmission power, the circles are equal.
- Random geometric graphs, formed as unit disk graphs with randomly generated disk centers, have also been used as a model of percolation and various other phenomena.



UDGs: Vertices and Edges

- The UDG is an abstract model of an ad hoc network.
  - It is a graph $G(V, E)$ with V the set of vertices and E the set of its edges.
  - V: Vertices are the sensor nodes. smart phones (UEs)
  - E: Edges between vertices represent connectivity, i.e., whether or not they can communicate.
- Why the name UDG (Unit Disk Graph)?

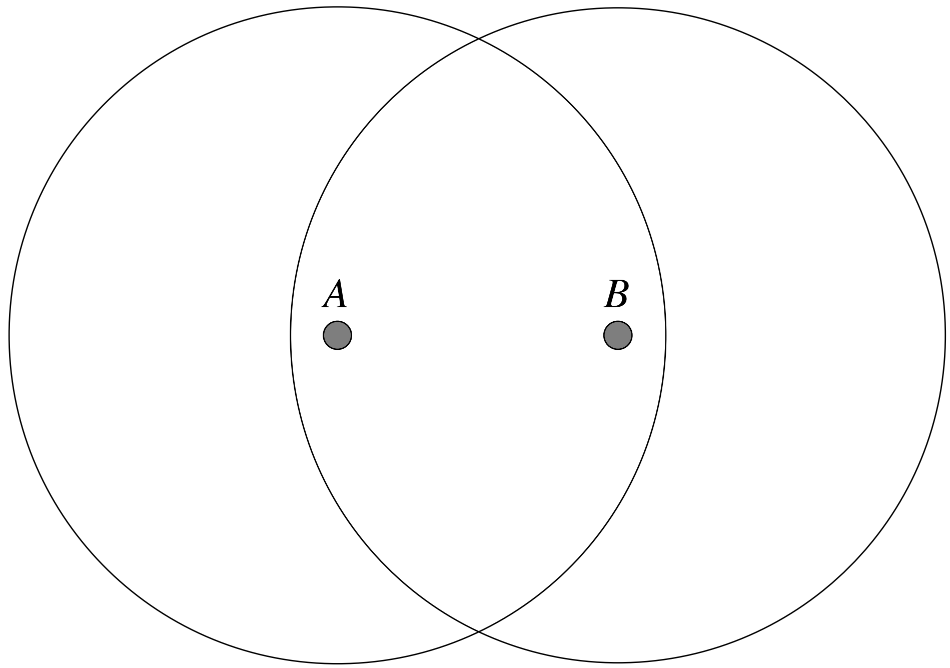
range



$$\{u,v\} \in E \iff d(u,v) \leq 1$$

Why UDG?

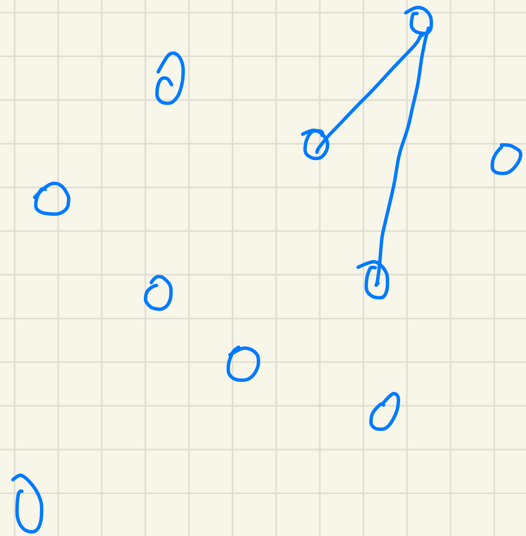
- Two (mobile) hosts A, B are adjacent if they are within reach of each other:



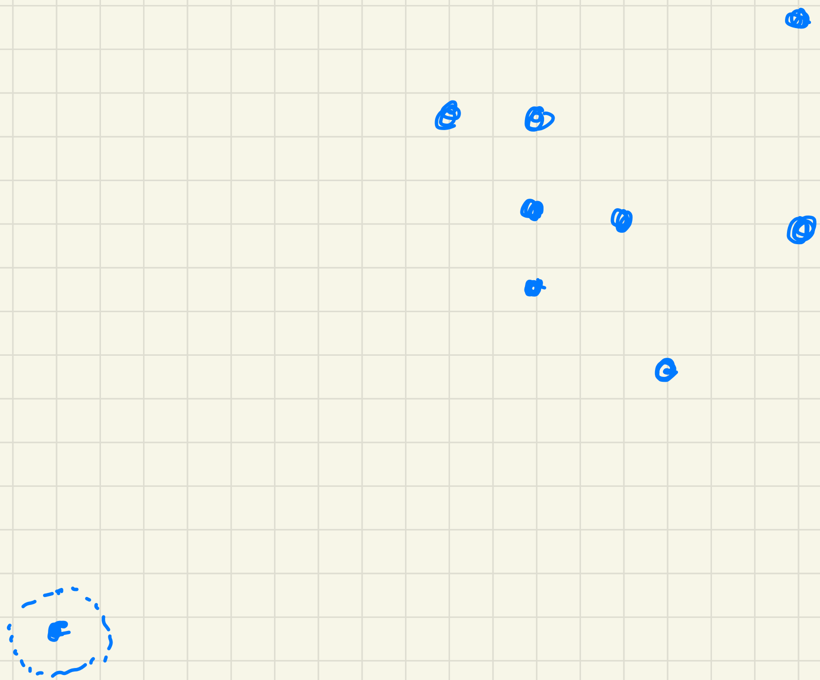
- There is an edge between A, B if and only if $d(A, B) \leq 1$.

1,
r

All "sensors" have equal range



$r = 1$

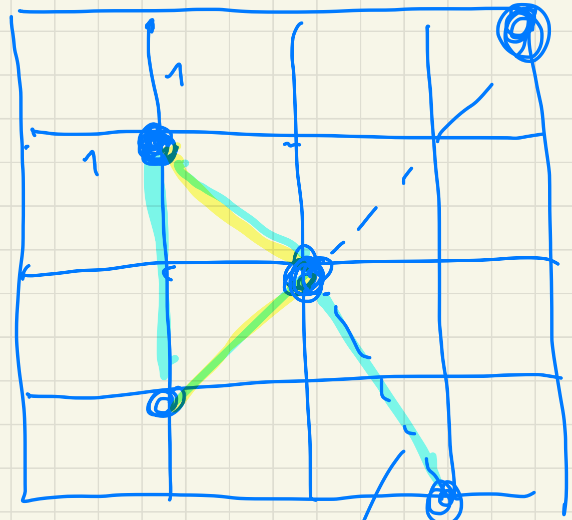


$r = 1$

$r = \sqrt{2}$

$r = 2.5$

$r = 3$

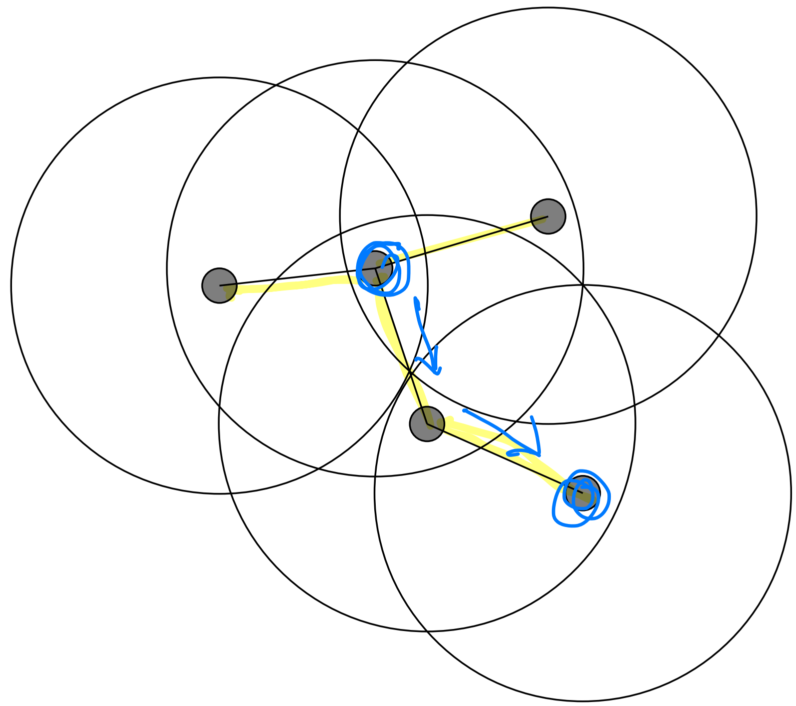


$2\sqrt{2}$

$\sqrt{5}$

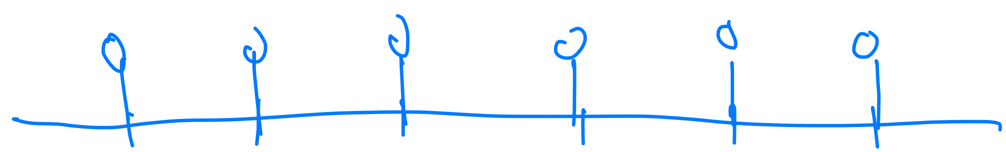
Example of UDG

- Underlying graph.



The transmission ranges of the sensors define a communication graph.

- The disks determine an "underlying" graph.



UDGs and Mobility

- The previous UDG model is static.
- If you want to include mobility then time t must be incorporated in the model.
- $G_0, G_1, \ldots, G_t, \ldots$ is a sequence of UDGs whereby G_t is the "state of the ad hoc network" at time t.
- Given G_t the new network G_{t+1} is obtained from G_t by the addition/deletion of nodes/links.

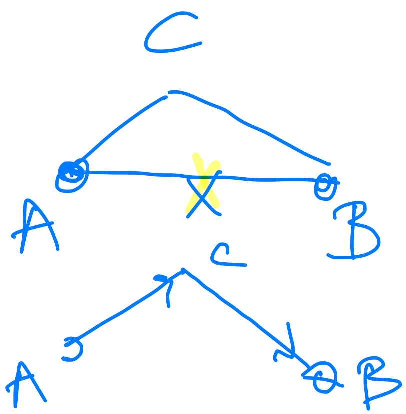
$G_0, G_1, G_2, \ldots$

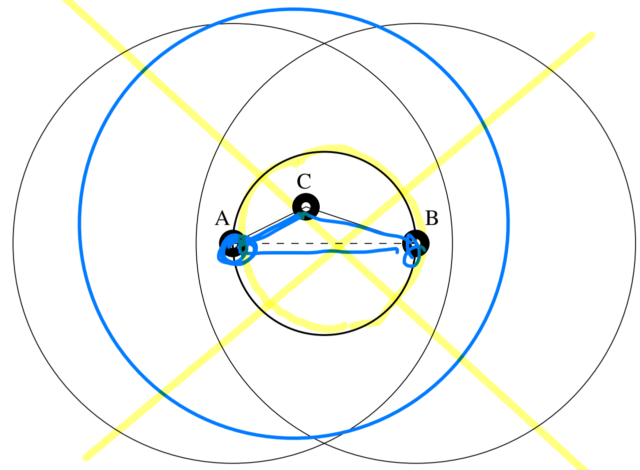
$G_i \longrightarrow G_{i+1}$

Gabriel Test

Gabriel Test (Algorithm)

- Assume points A and B are within range of each other.



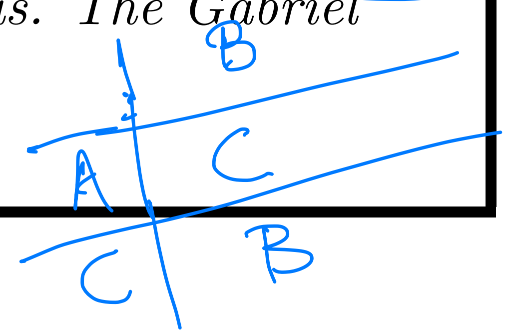


A, B, C determine disk

1. Draw circle with diameter AB.
2. If there is another point, say C inside this circle then remove the link connecting A to B (is not needed!)

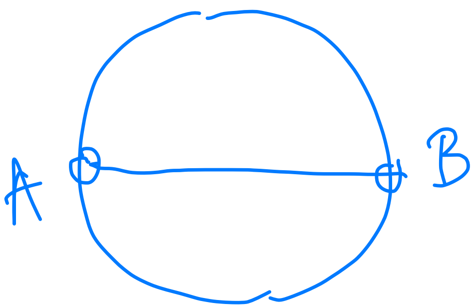
Routing Table

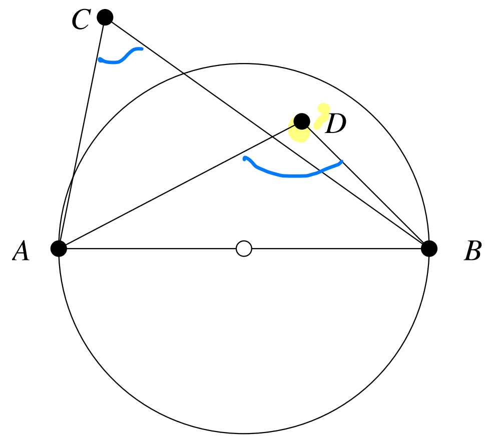
- **Theorem 1** *Assume a connected wireless network with node ranges represented as circles of identical radius. The Gabriel algorithm removes all edge crossings!*



Gabriel Graph: Observations

- Call AB a Gabriel edge if the circle with diamater AB contains no other points.





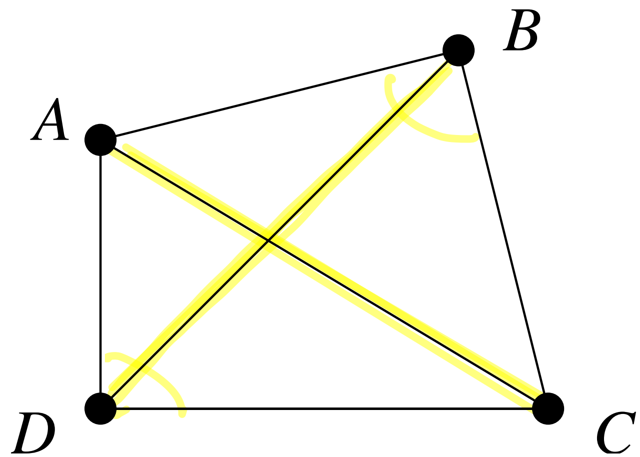
Gabriel edges are essential!

- A point X is inside the circle with diameter AB if and only if the angle AXB is bigger than $\pi/2$.

- A point X is inside the circle with diameter AB if and only if its distance from the center of the circle is bigger than $|AB|/2$.

Gabriel Graph is Planar

- If the Gabriel edges AC and BD intersect, A, B, C, D must form a convex quadragon!

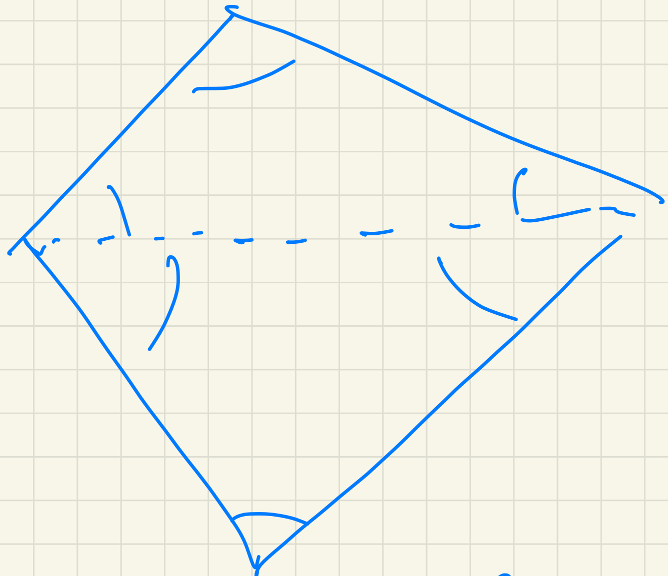


$\measuredangle B > \pi/2$
$\measuredangle D > \pi/2$
$\measuredangle A > \pi/2$
$\measuredangle C > \pi/2$

sum $> 2\pi$

AC is Gabriel edge because there is no point inside circle of diameter $|AC|$

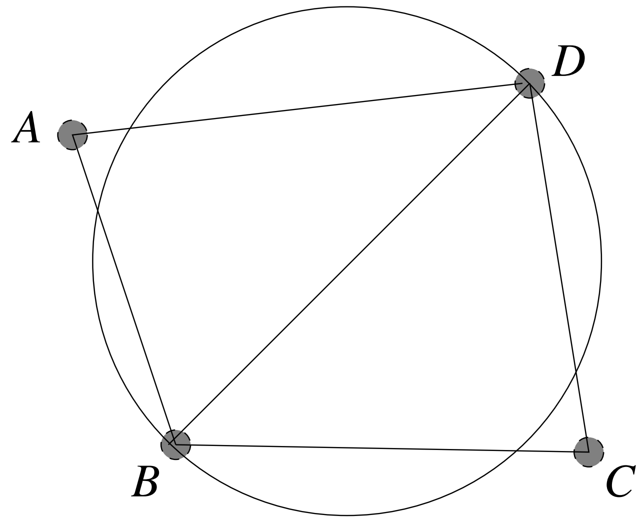
BD is a Gabriel edge because there is no point inside circle of diameter $|BD|$



sum of angles is 2π

Why is edge BD preserved?

- Edge BD is preserved because it is a Gabriel edge.

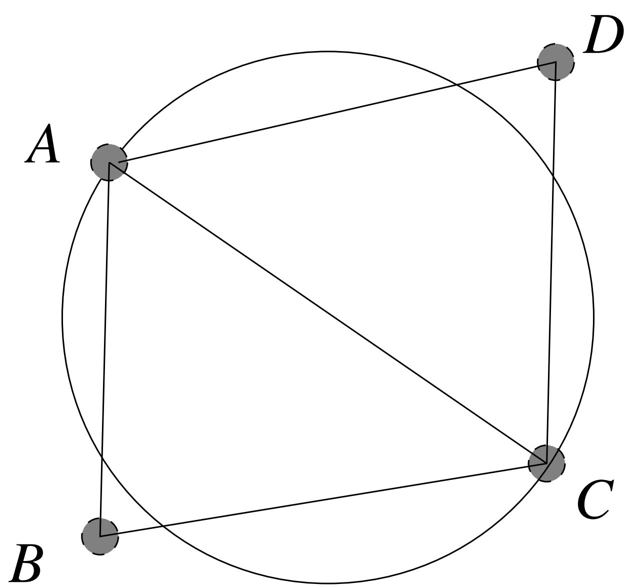


- Therefore both A and C lie outside the circle with diameter BD.

Repetition

Why is edge AC preserved?

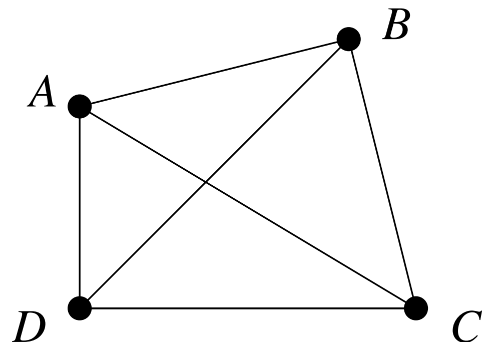
- Edge AC is preserved because it is a Gabriel edge.



- Therefore both B and D lie outside the circle with diameter AC.

Gabriel Test Removes Edge Crossings

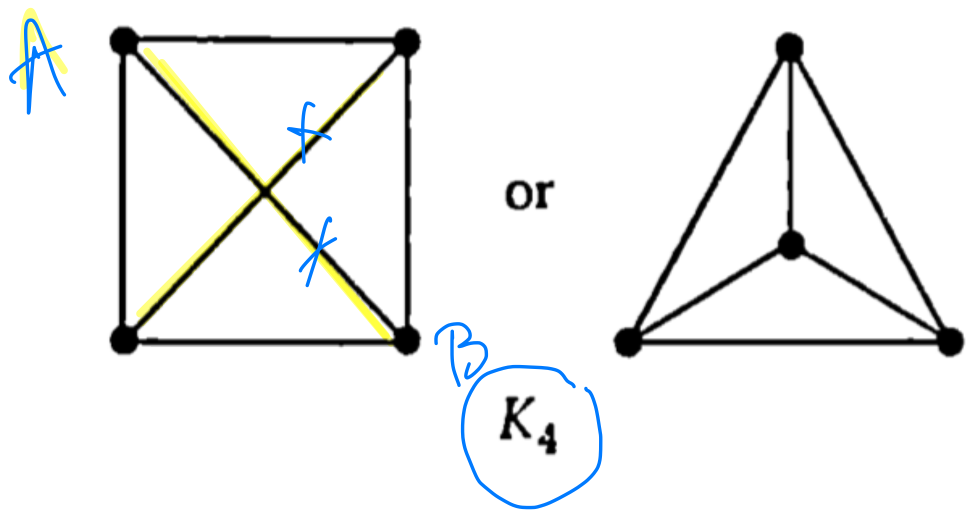
- If the Gabriel edges AC and BD intersect, A, B, C, D must form a convex quadragon!



- Hence $\angle ABC, \angle BCD, \angle CDA, \angle DAB < \pi/2$, contradicting the fact that $\angle ABC + \angle BCD + \angle CDA + \angle DAB = 2\pi$.

Examples: Gabriel Graph Depends on the Drawing!

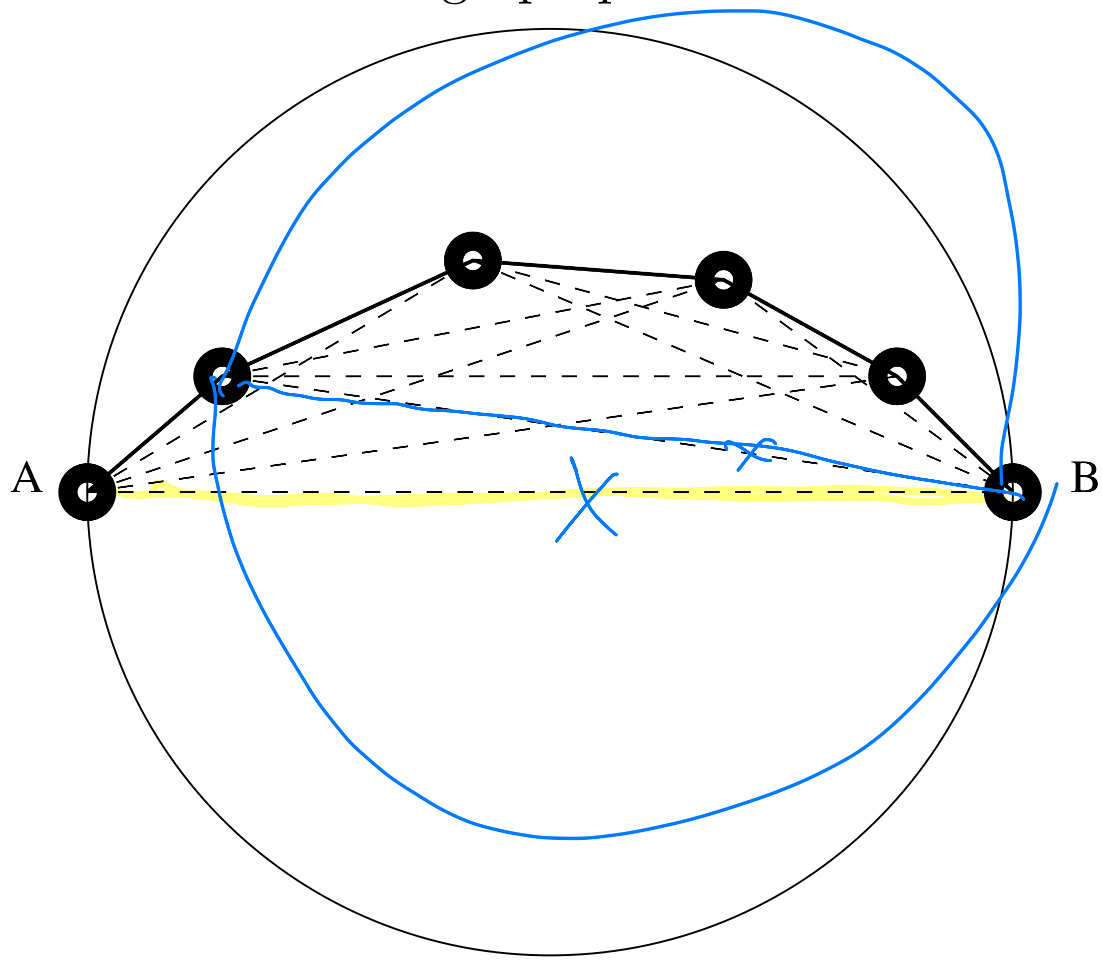
- Find the Gabriel graph in each case:



crossings no-crossings

Example: Gabriel Test and Shortest Paths

- How well does the Gabriel graph perform?



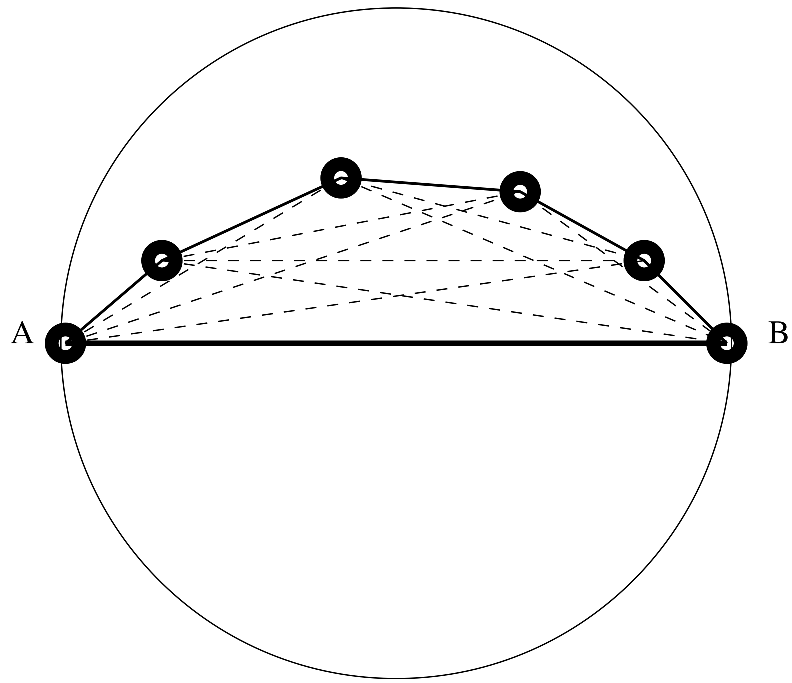
- For more details, see Appendix

How about Deleted Edges?

- You maintain a *Routing Table*.
  - A data base that when you are at A you ask:
    How do I reach B?
  - It gives you the answer: Go to C.
  - And when you reach C you ask again:
    How do I reach B?
  - It gives you the answer: Go to B.
- Standard routing table contains an entry for each possible destination with the out-going link to use for destination
- Message delivery proceeds in the obvious manner one link at a time, looking up the next link in the table.

Too many hops spoil the batteries!

The Gabriel test creates a planar graph but removes long links.



I could have reached B directly from A in one hop.

Instead it takes me five hops!

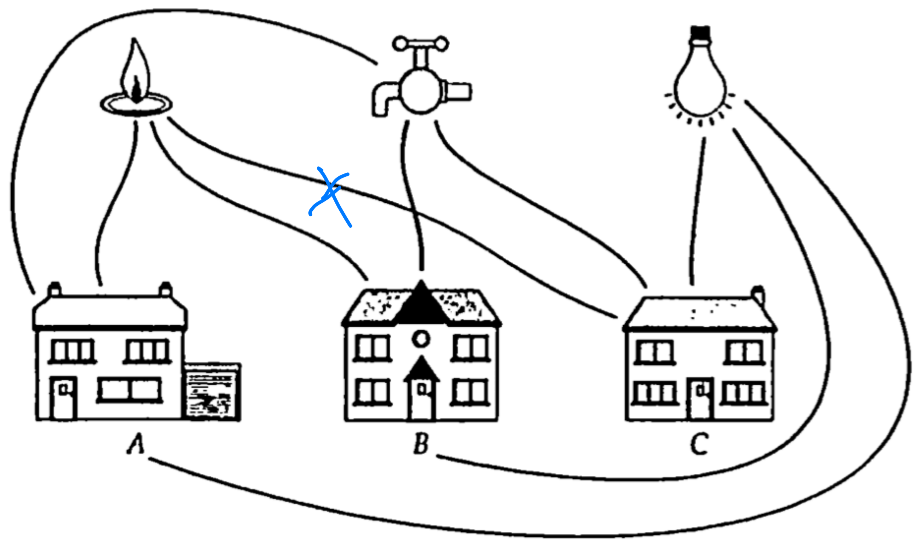
What is important: the Gabriel test will remove all crossings!

Planarity

- Planar Graph
  - A graph G is planar if it can be drawn in the plane in such a way that no two edges meet except at a vertex with which they are both incident.
  - Any such drawing is a plane drawing of G.
  - A graph G is non-planar if no plane drawing of G exists.
- The Gabriel test produces a planar network!
  - It was done by removing edges!

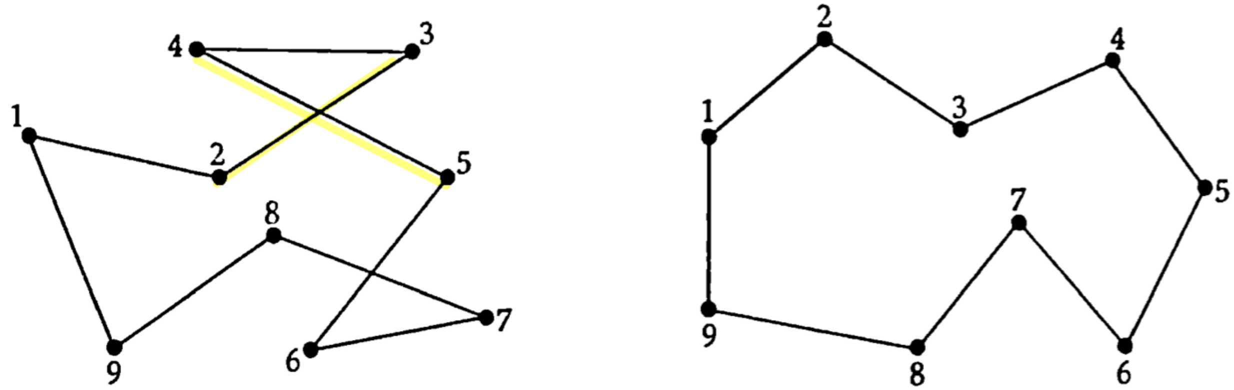
Planarity: Sometimes it is not Possible

- Impossible to draw as a planar graph

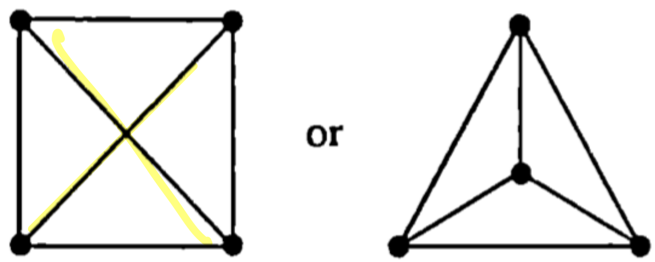


Planarity: Depends on the Drawing

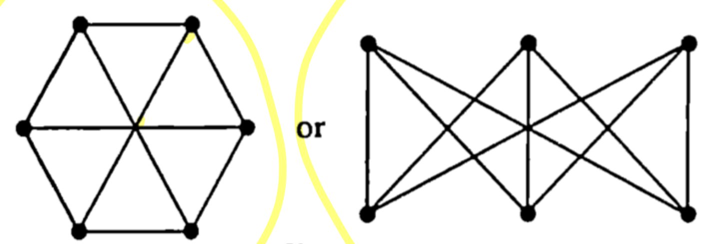
- Just redraw the graph:



- Just redraw the graph:



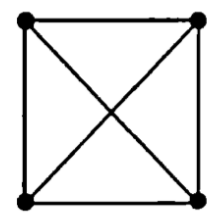
K_4



$K_{3,3}$

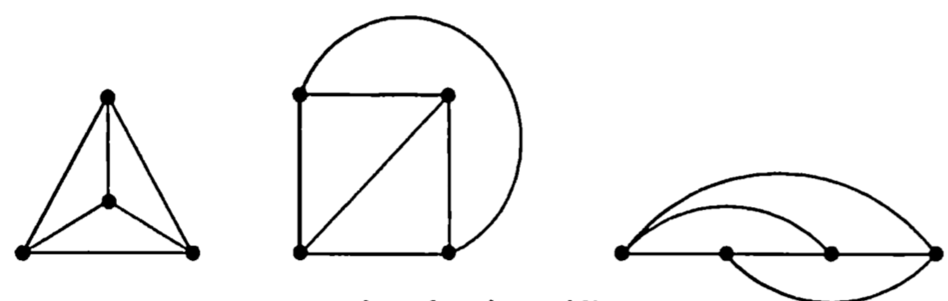
Planarity: Depends on the Drawing

- K_4



K_4

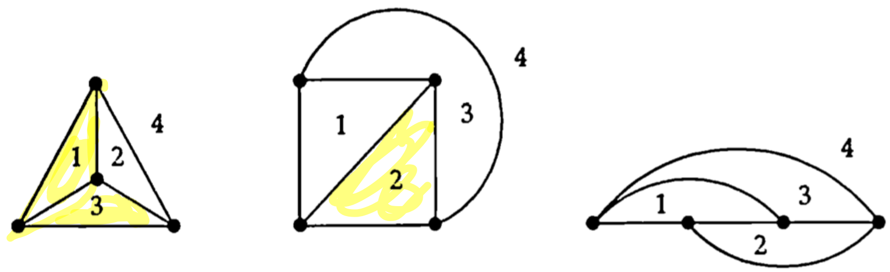
- Different ways to draw K_4



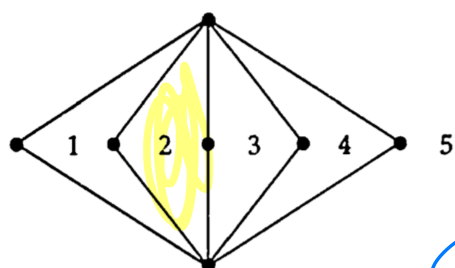
plane drawings of K_4

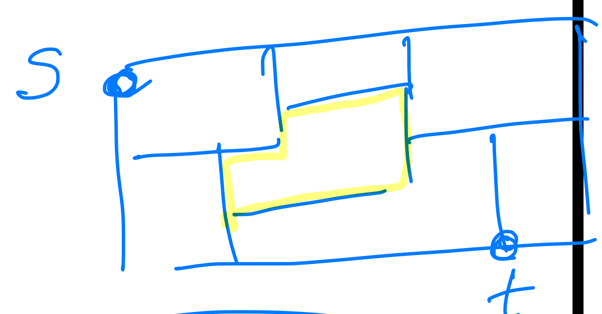
Faces of a Planar Graph (1/2)

- Every plane drawing of a planar graph divides the plane into a number of regions.
- For example, any plane drawing of K_4 divides the plane into four regions: three triangles (3-cycles) and one *infinite region*



- Another example

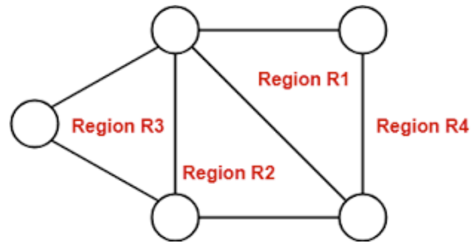




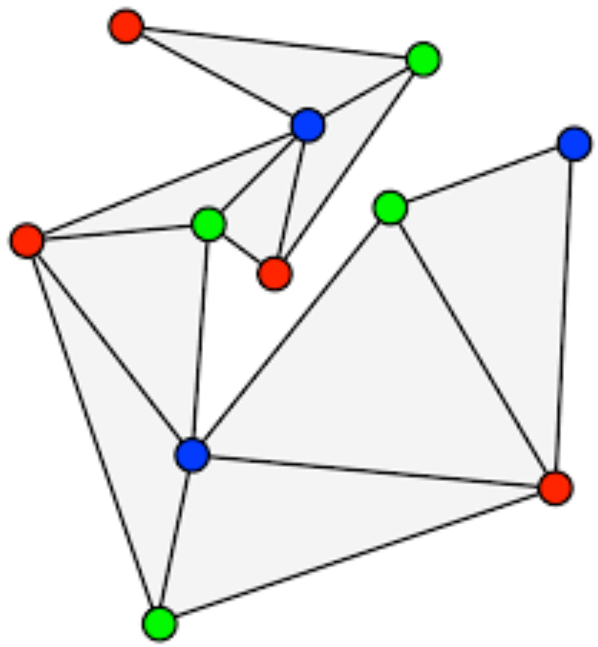
Face Routing

Faces of a Planar Graph (2/2)

- What are the faces of the planar graph?



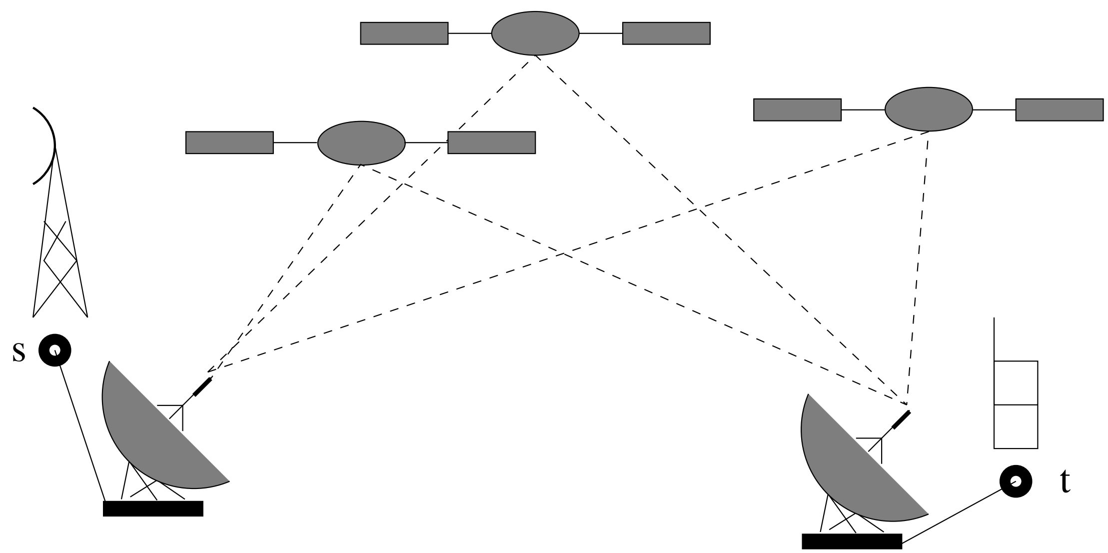
- What are the faces of the planar graph?



Geometric Routing

Global Positioning Systems (GPS)

Can use GPS to discover the (x, y) coordinates of the target node.



GPS uses three satellites in line of sight.

It determines location by **time-of-arrival** differences (temporal delays of several signals).

Can always construct an undelying geometric planar graph using the Gabriel test!

Routing in a Geometric Planar Network

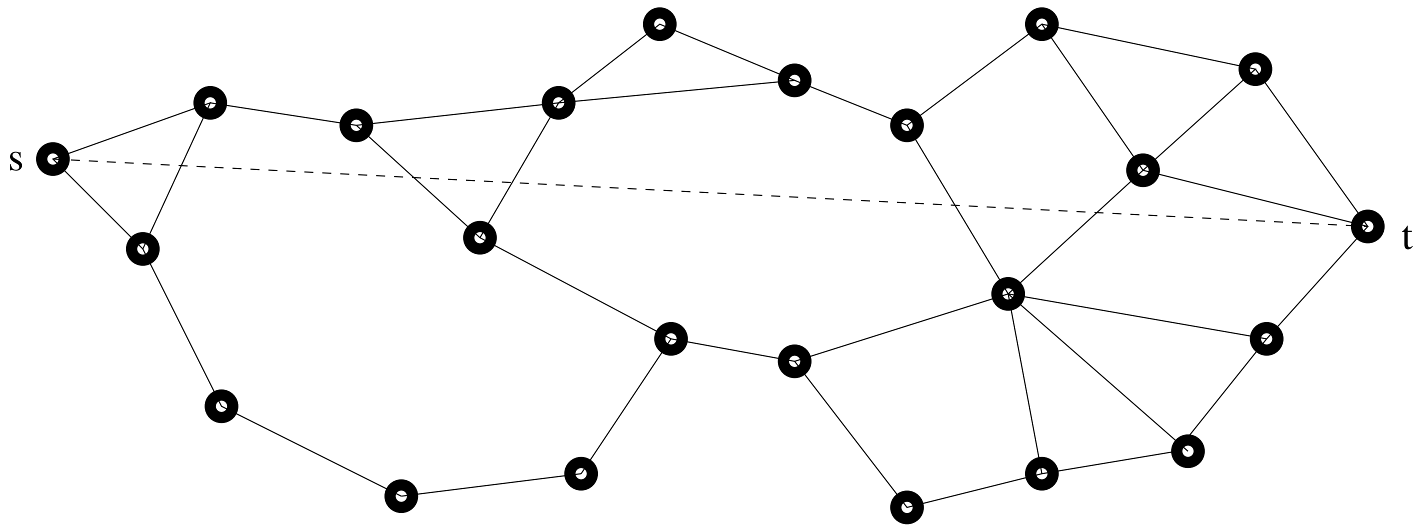
Input: A geometric graph.

Goal. Go from source node s to target node t.

- We need some kind of "capabilities" in order to move towards the target t. This may include the following
  - Updating coordinates of current position c.
  - Must know the coordinates of t.
  - If c is our current position we need to be able to determine the slope of the line $\vec{ct}$.
- We need to be able to determine the slopes of the edges incident to our current position.

Back to Route Discovery

After applying the Gabriel test we have a planar graph.



Using GPS we can find out the (x, y) coordinates of s and t.

Hence, we can compute the slope of the line $\vec{st}$.

But how do we use this information in order to discover a route?

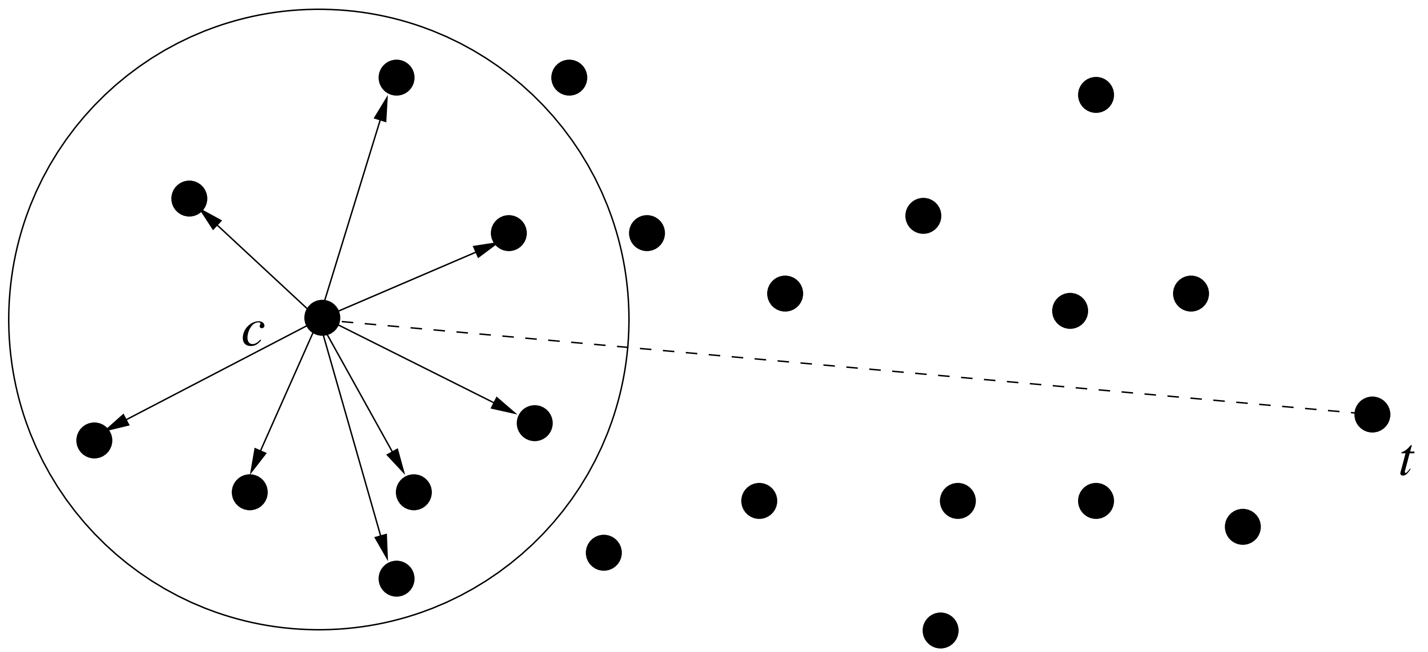
Compass Routing Algorithm

1. Start at source node $c := s$.
2. in a recursive way:
   (a) Choose edge of our geometric graph incident to our current position and with the smallest slope to that of the line $\vec{ct}$.
   (b) Traverse the chosen edge.
   (c) Go back to (a) and repeat until target t is found

- **Theorem 2** *Compass routing requires GPS and works in many cases (like, random graphs with high probability) and is the basis of tiny OS.*

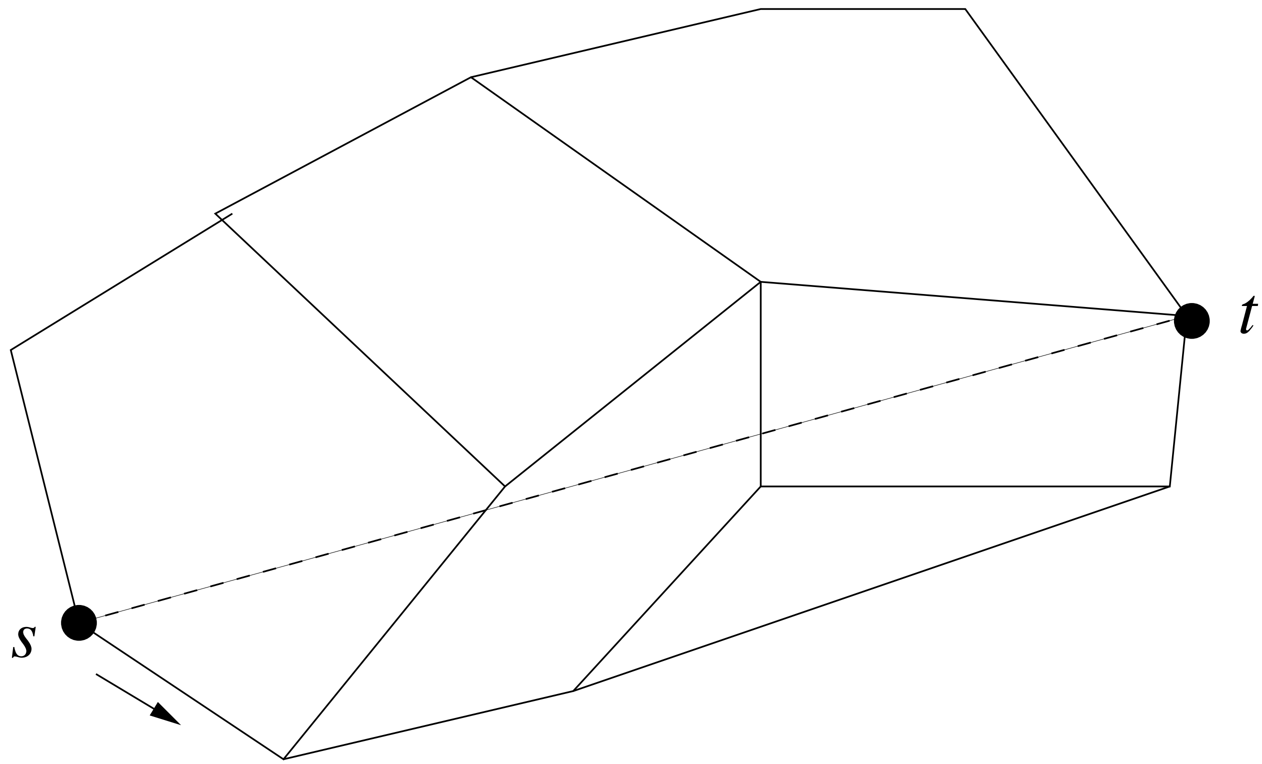
Compass Routing: Next Move

- Choose the smallest angle!

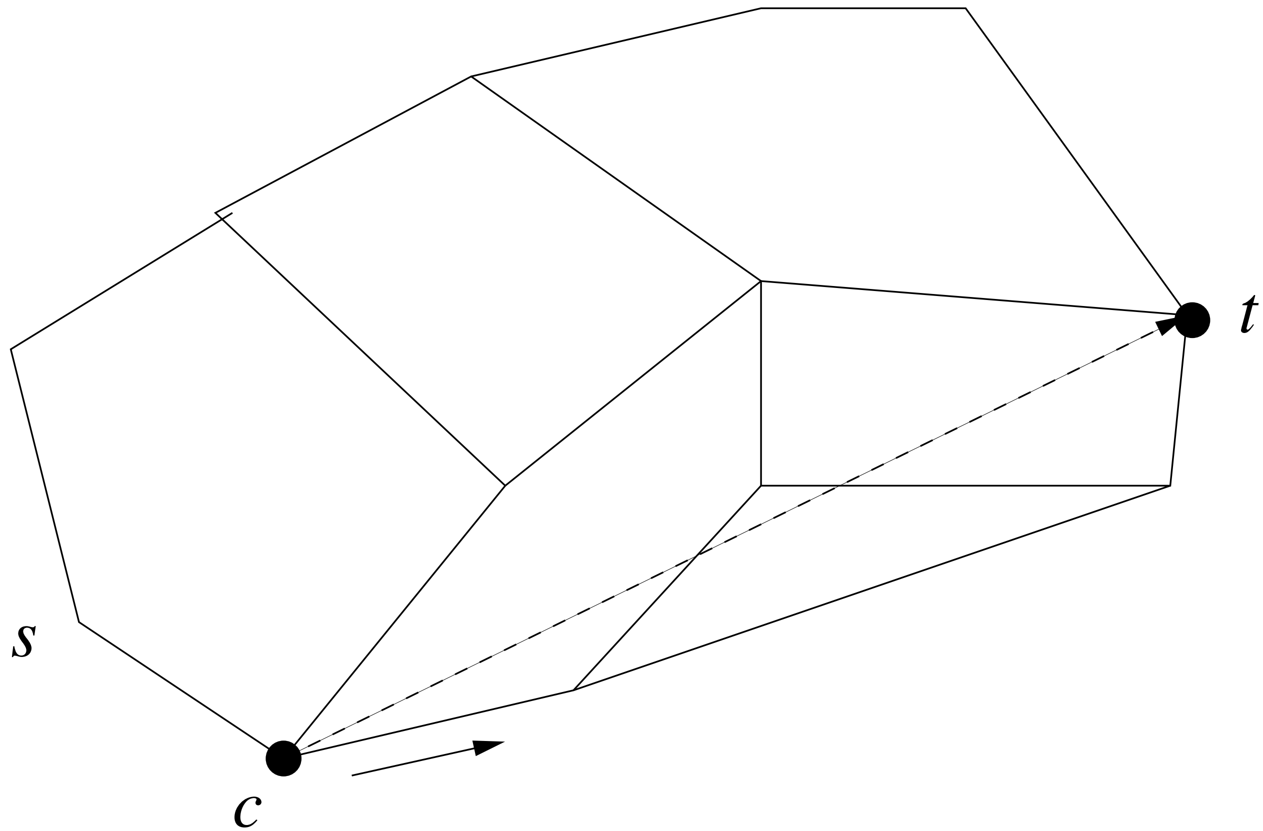


- **Problem:** Compass routing can fail to reach destination!

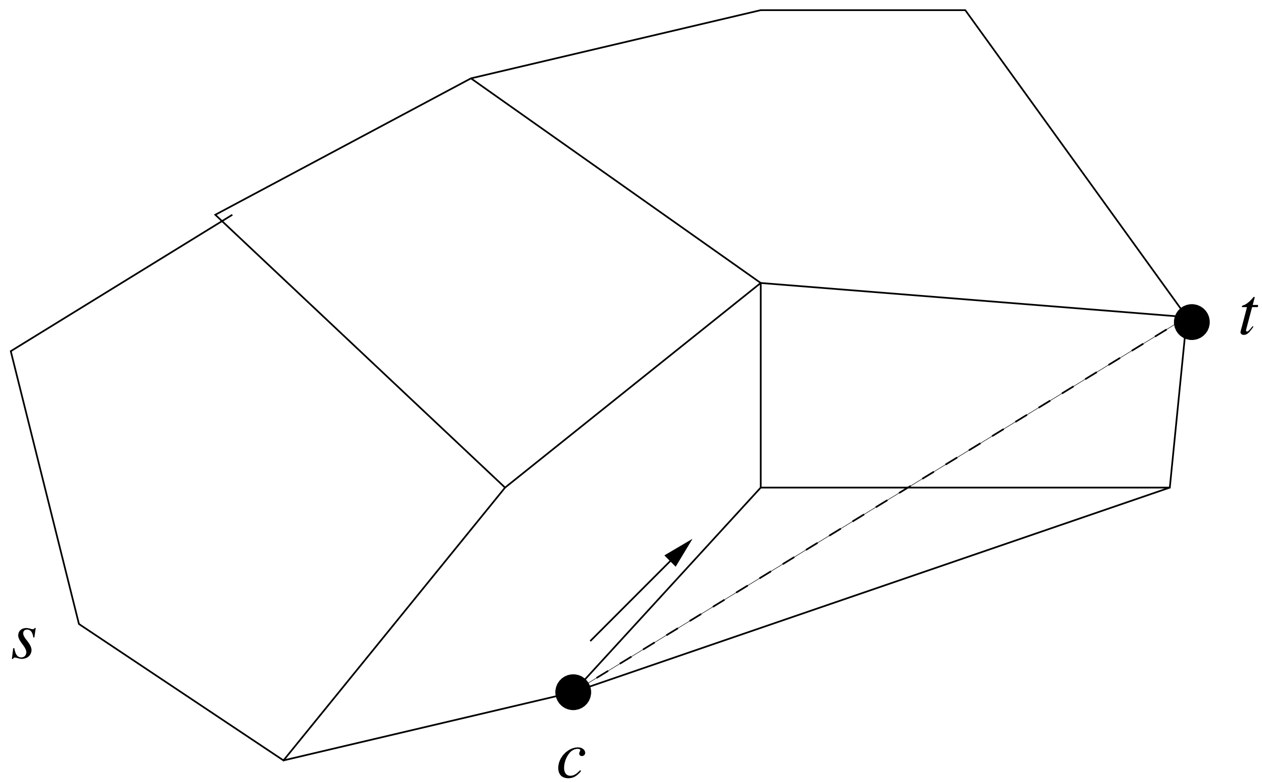
Compass Routing (1/5)



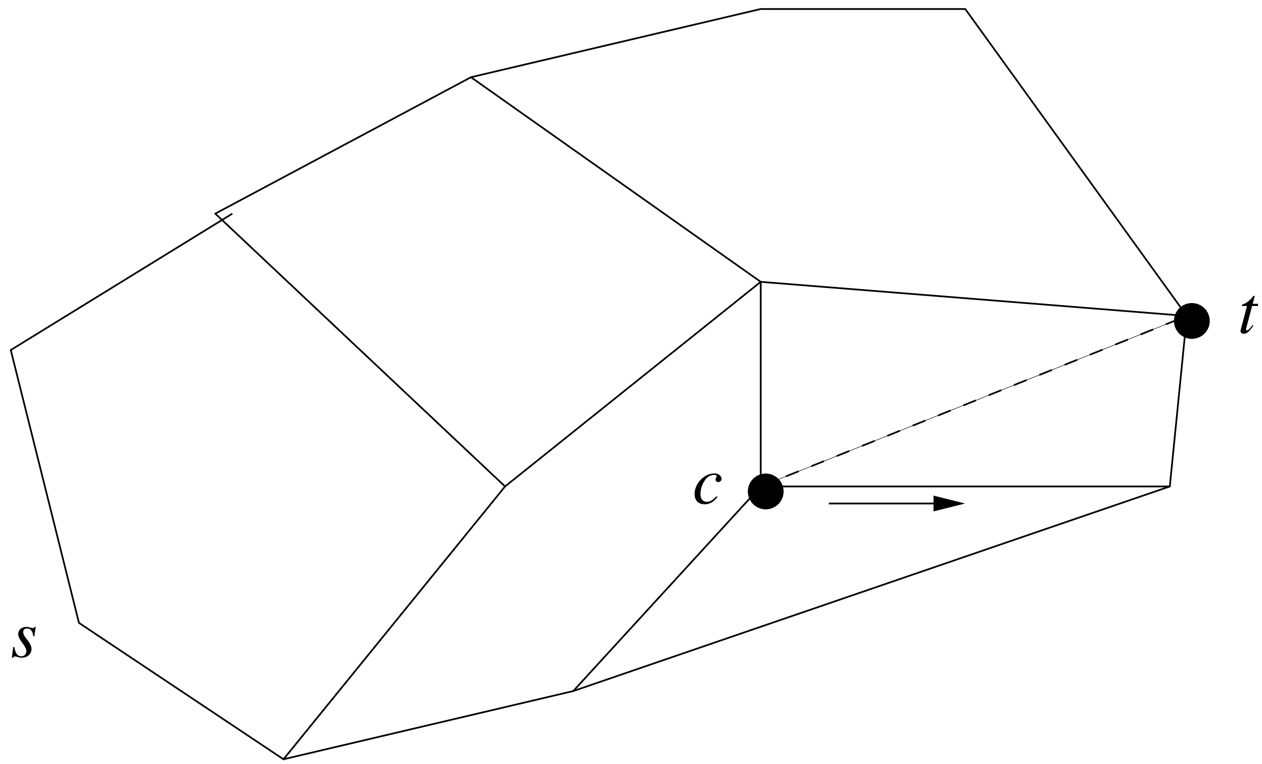
Compass Routing (2/5)



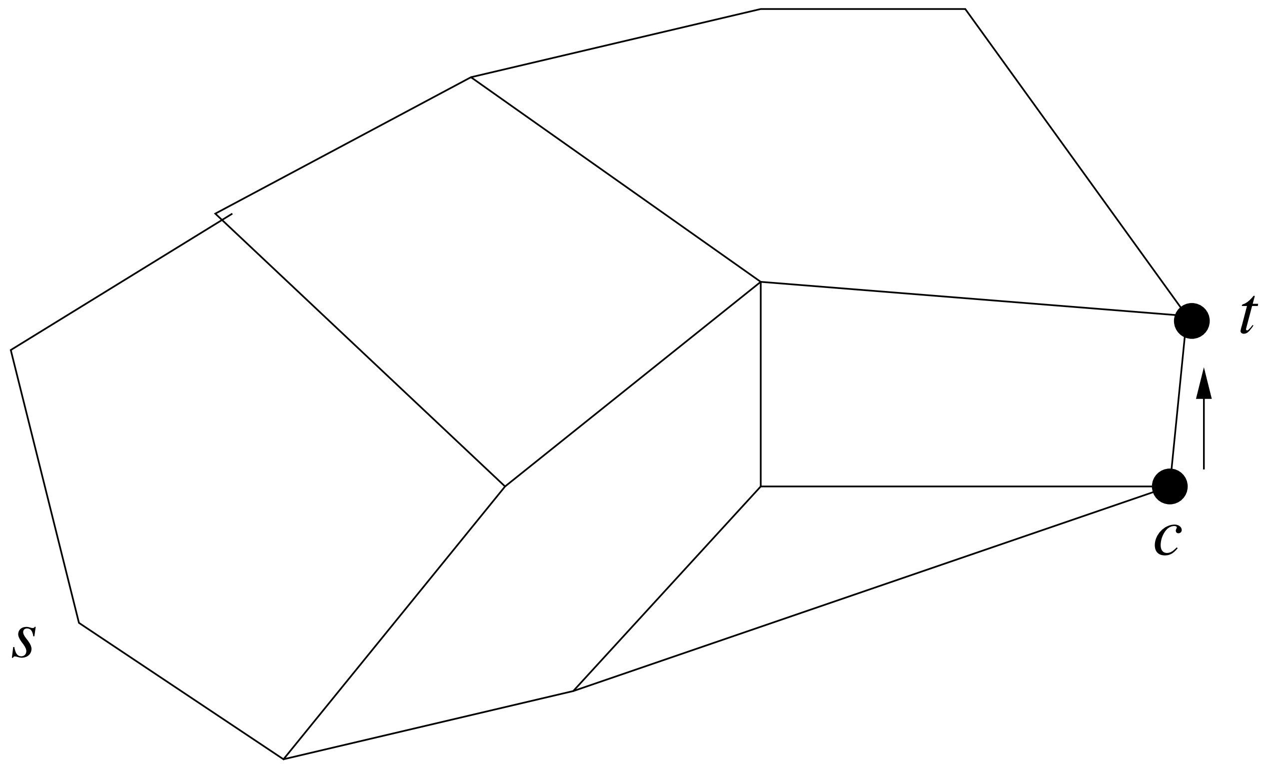
Compass Routing (3/5)



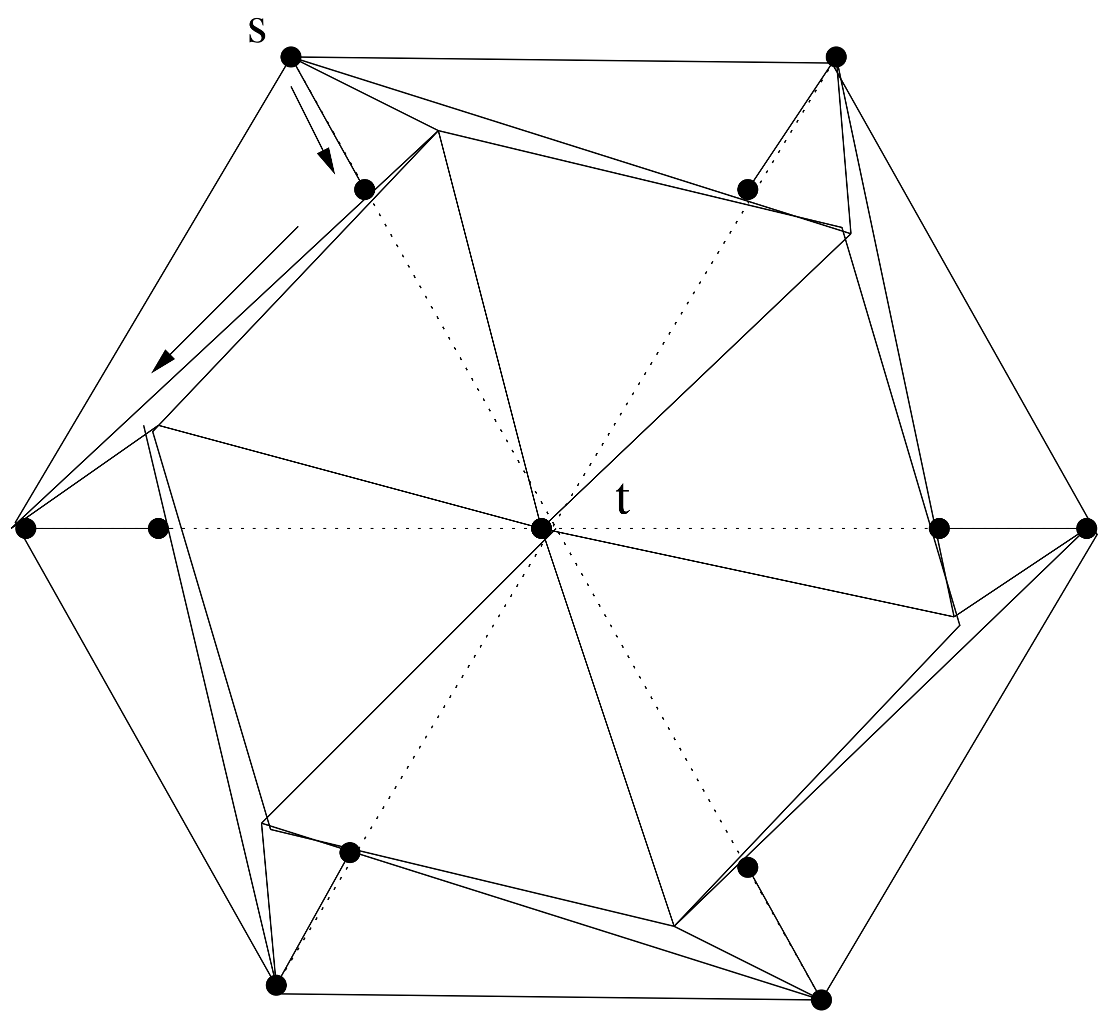
Compass Routing (4/5)



Compass Routing (5/5)



Compass Routing May not always Work!



Face-Routing Algorithm (1/3)

- Starting Phase
  1. Let s, t be the source and target nodes in a geometric graph.
  2. Determine the straight line $\vec{st}$ and remember it.
  3. Start with $c := s$ as the current node.
- Note that one must remember the straight line $\vec{st}$, which remains the same throughout the algorithm.

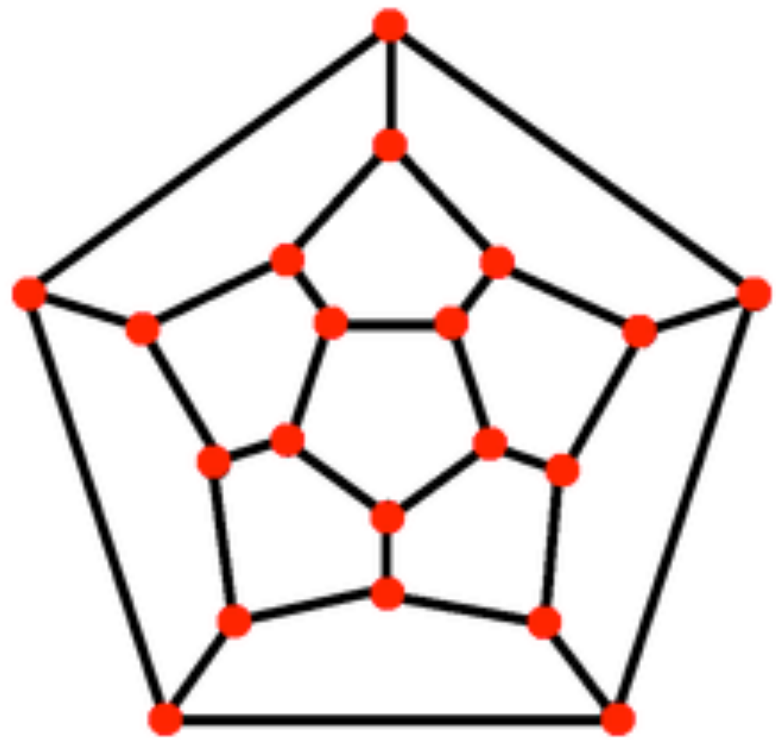
Face-Routing Algorithm (2/3)

- Face selection and traversal phase:
  1. Determine the face F incident to c such that F intersects the straight line $\vec{st}$. This determines an edge one of whose endpoints is c.
  2. Select a direction of movement (Left or Right) and move along the edges of the face F.
  3. In this traversal, eventually you hit an edge, say $\{u, v\}$, which crosses the straight line $\vec{st}$. If neither u nor v is equal to t then select the first vertex u and update the current vertex $c \leftarrow u$.
  4. Iterate: Go back to Item 1.
- Notice that you have the choice to go either Left or Right. It does not matther which direction you select.

Face-Routing Algorithm (3/3)

- Final phase:
  1. Stop when t is found.
- Why does the algorithm terminate correctly?
- **Theorem 3** *Face routing requires GPS and works in all planar graphs (and is the basis of route discovery in many ad hoc networks).*

Example: Go from s to t

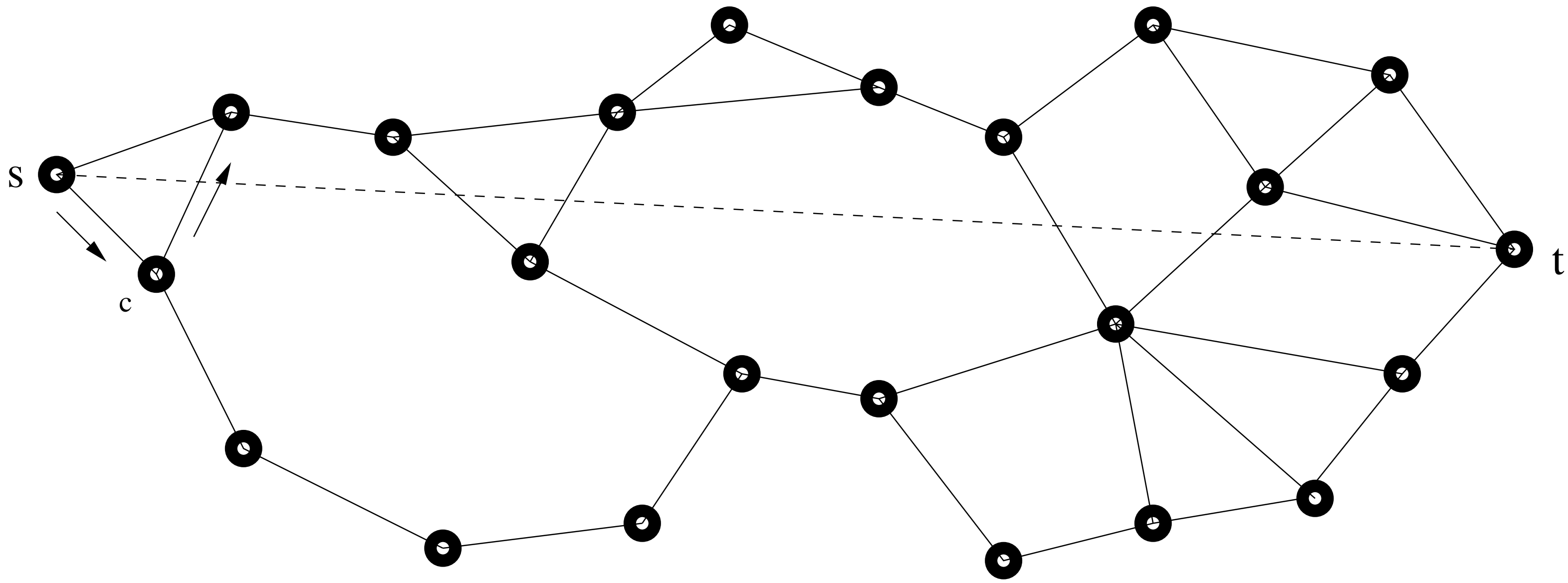


Initially $c := s$.

Update c and repeat.

If the last last edge you cross in a round is (a, b) then in the next round start from b

Example: Go from s to t



Initially $c := s$.

Update c and repeat.

If the last last edge you cross in a round is (a, b) then in the next round start from b

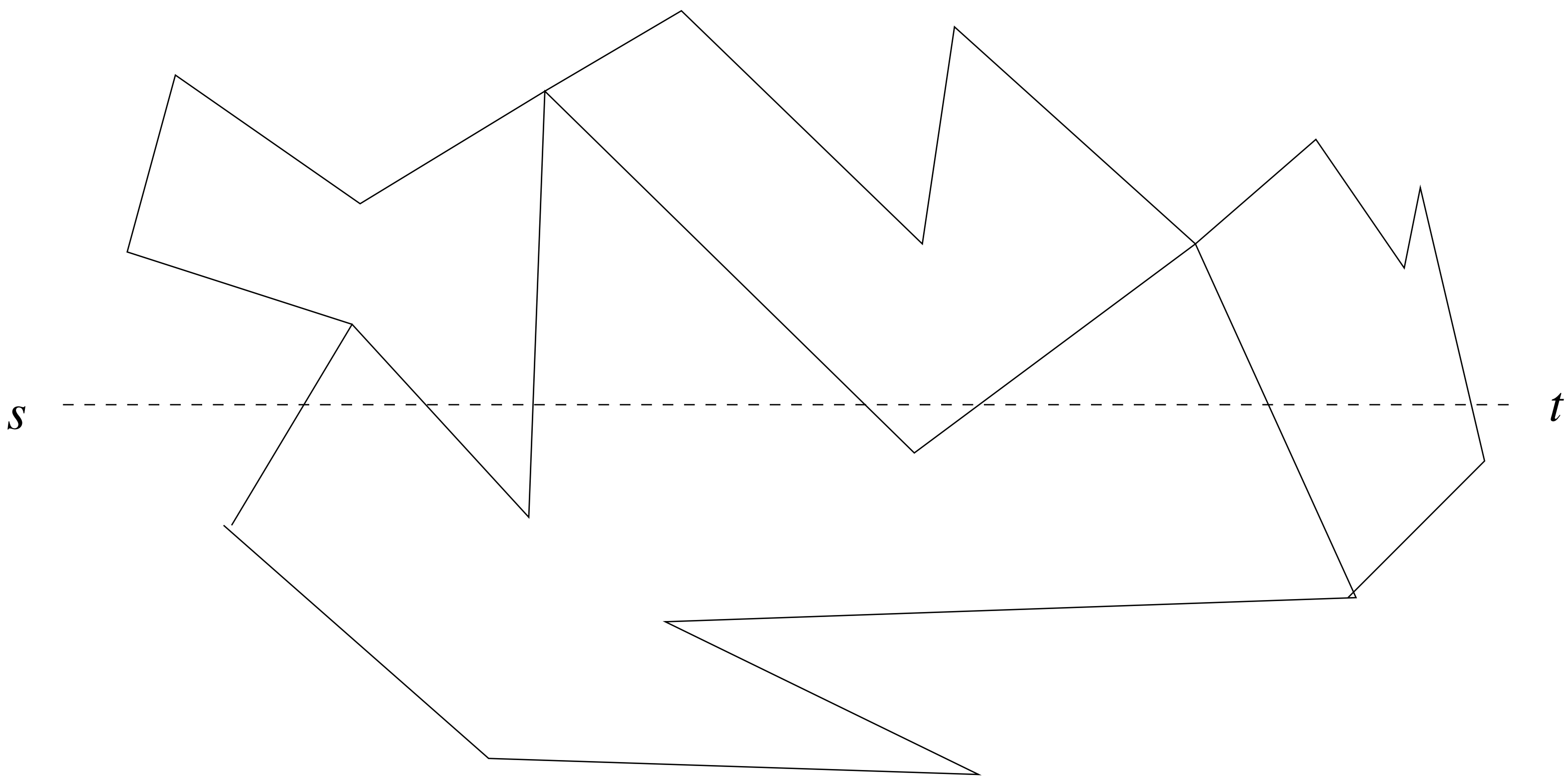
Analysis of Face-Routing

- Face routing always advances to a new face. We never traverse the same face twice.
- The distance from the current position c to t gets smaller with each iteration.
- Each link is traversed a constant number of times. Since the graph is planar face routing traverses at most $O(n)$ edges.

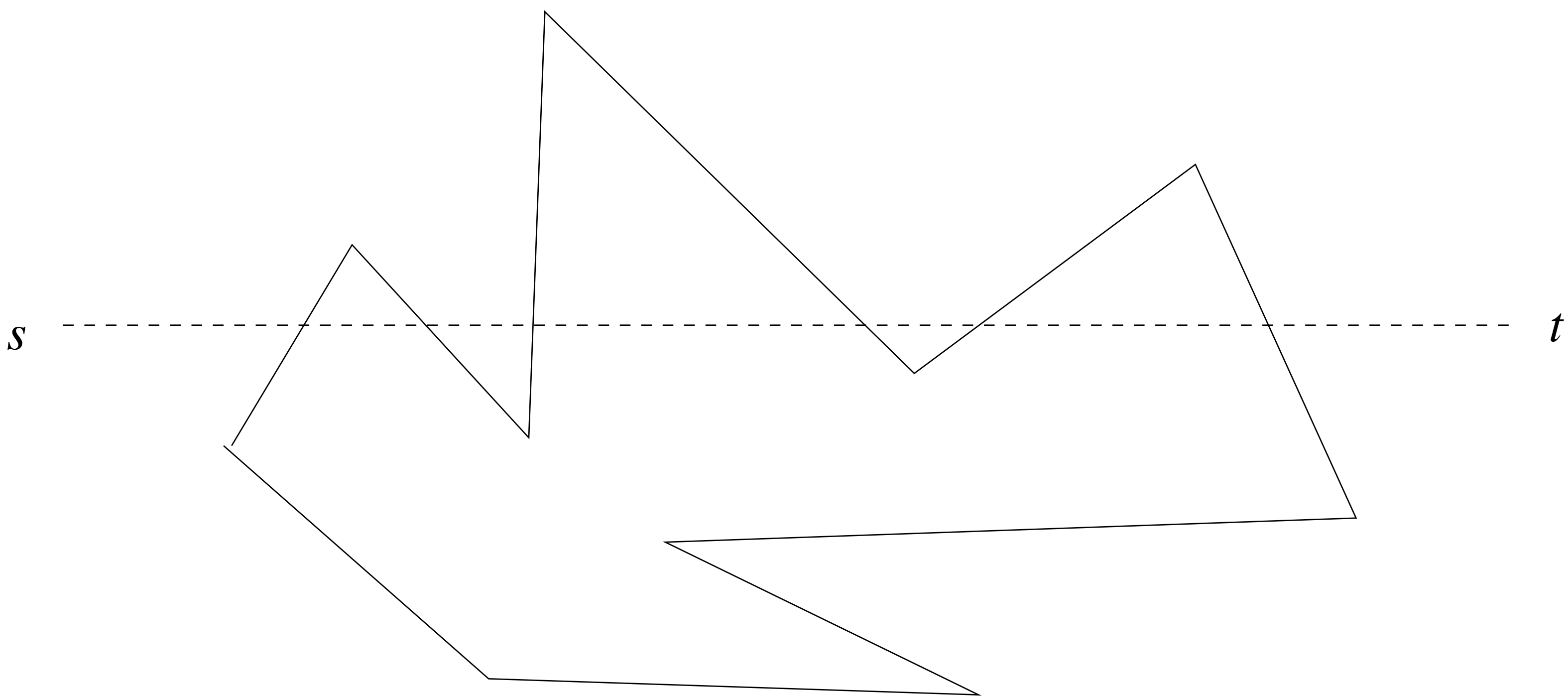
Problems with Face-Routing

- No indication how long is the Euclidean distance traveled!
- But does it matter? All we wanted was to discover a route!

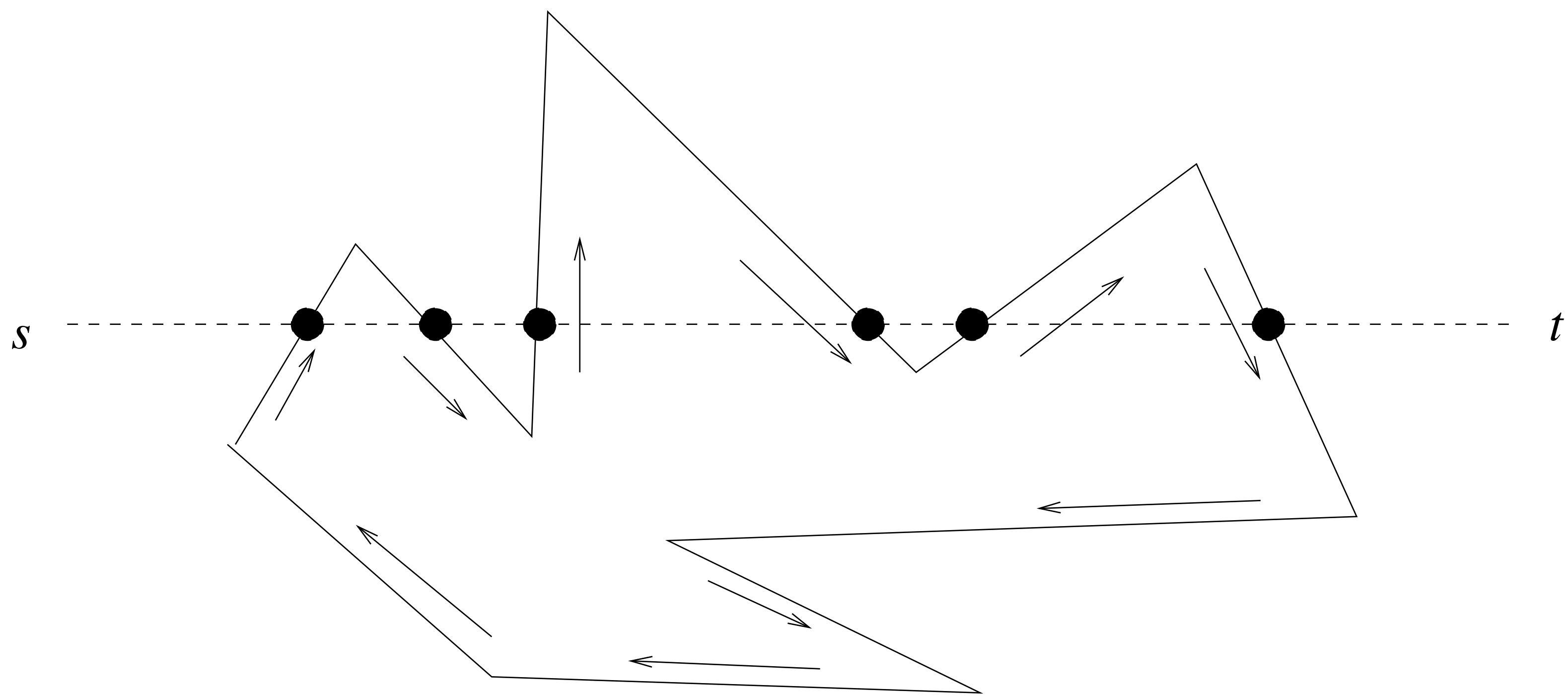
Example 1: How do you Traverse Faces?



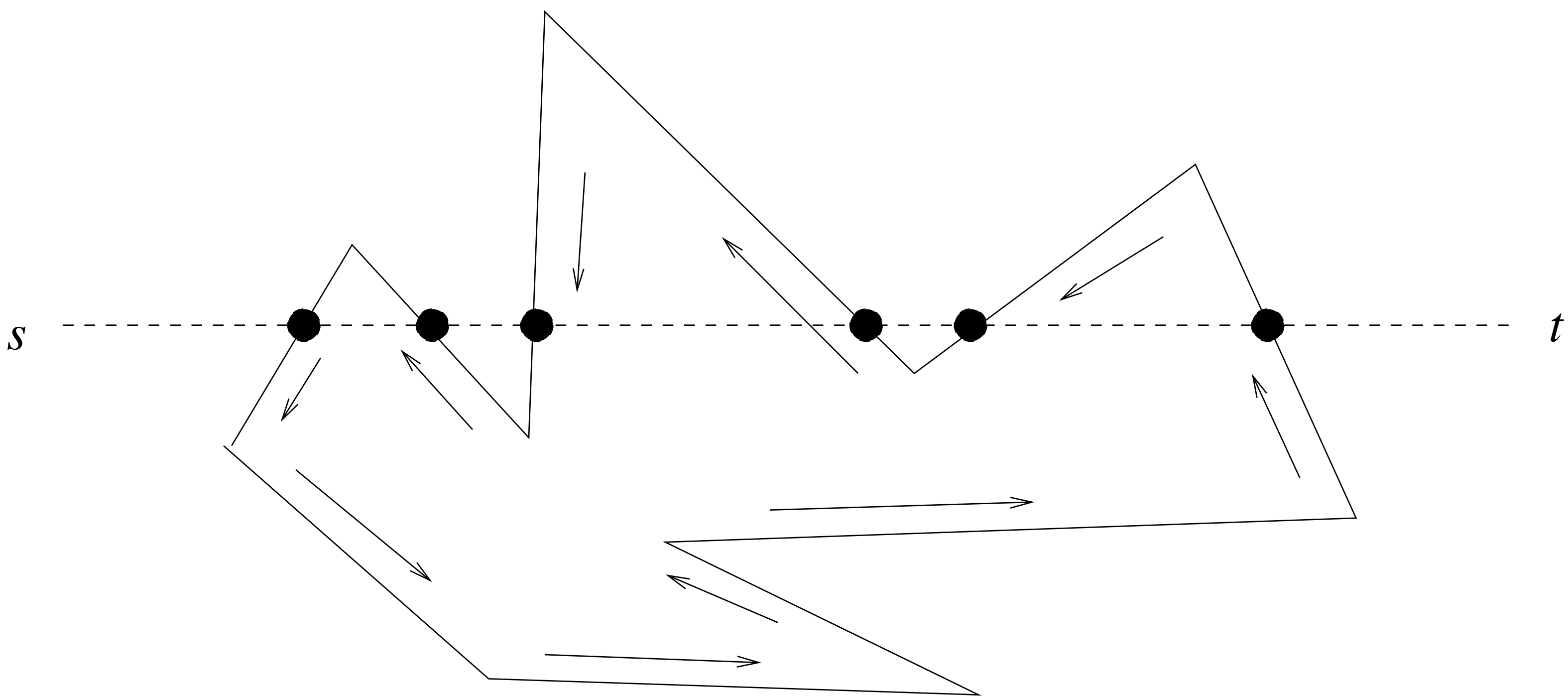
Example 1: When do you Flip Face?



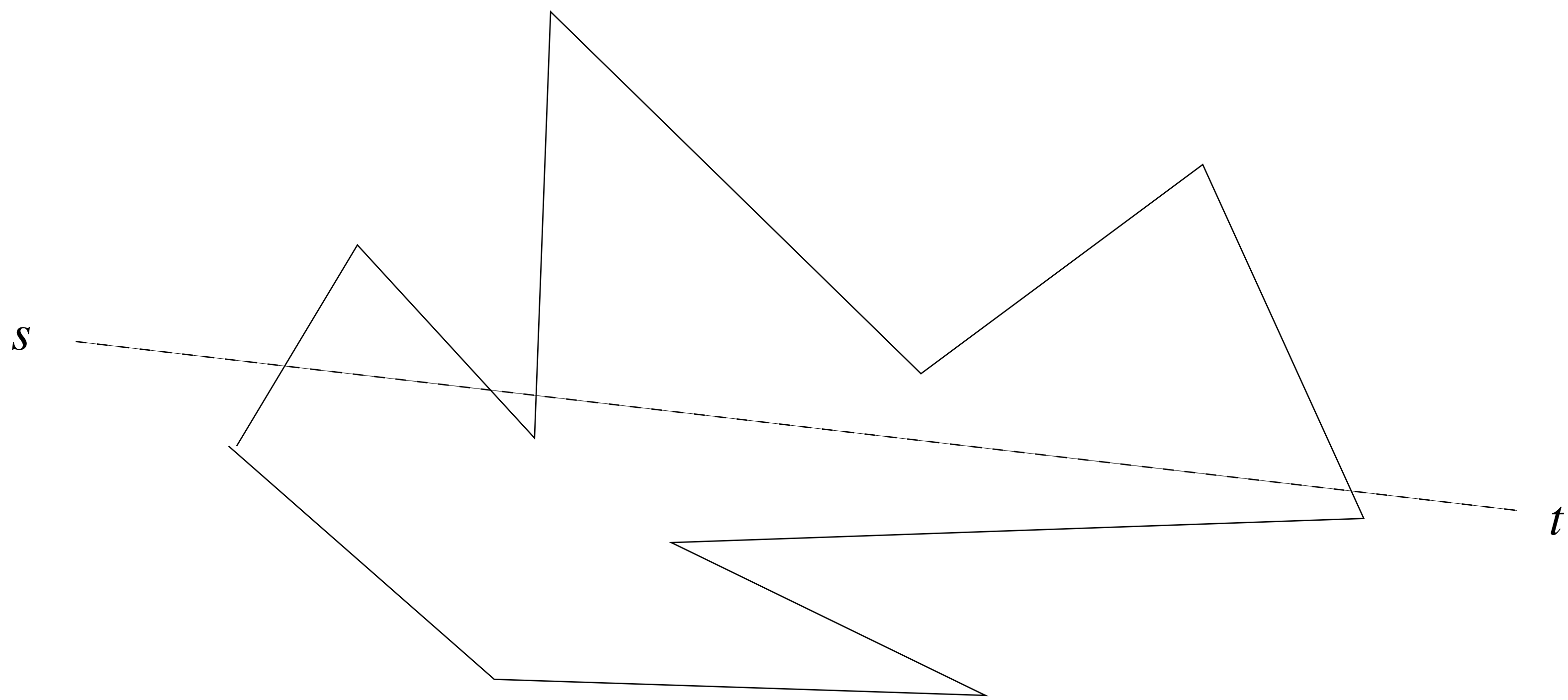
Example 1: Go to the Farthest Crosssings!



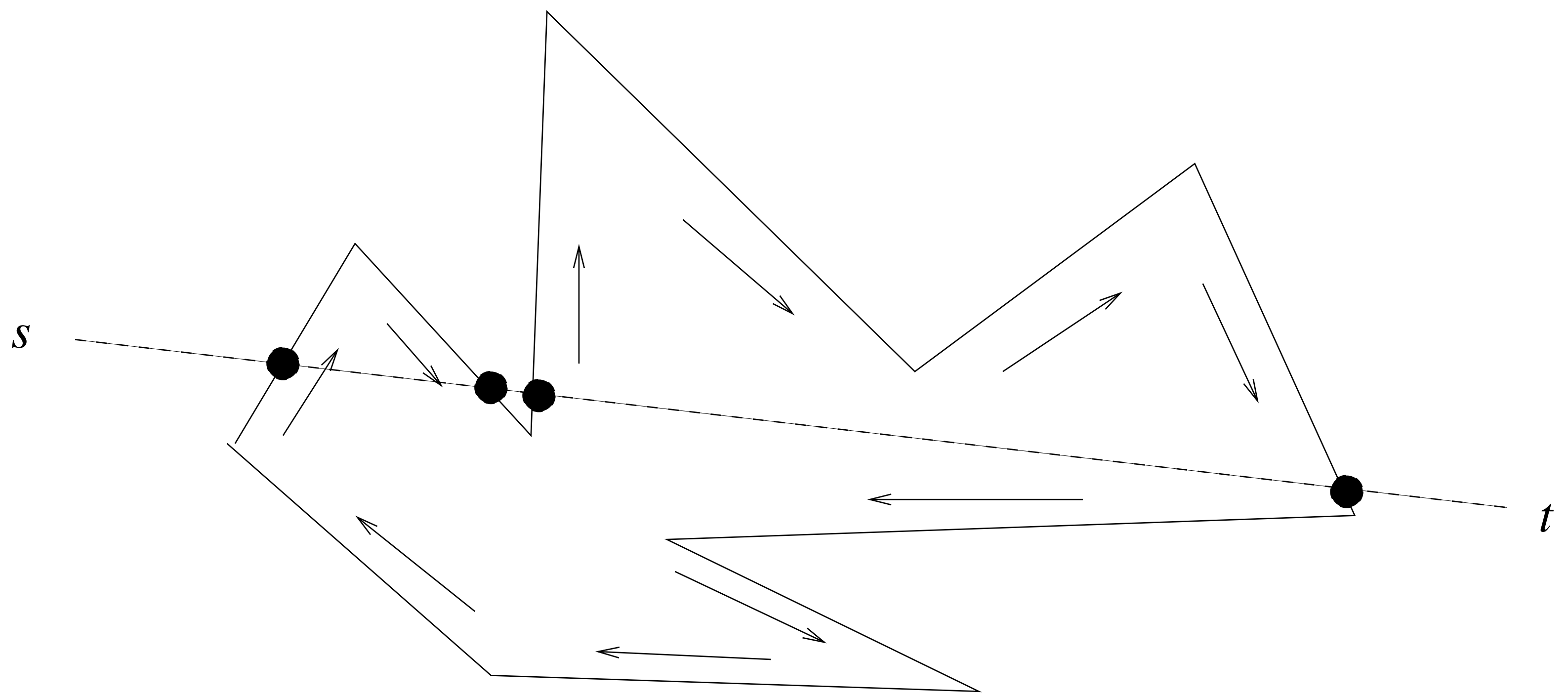
Example 1: Can Choose Right Hand Rule!



Example 2: Different Target, Different Direction?

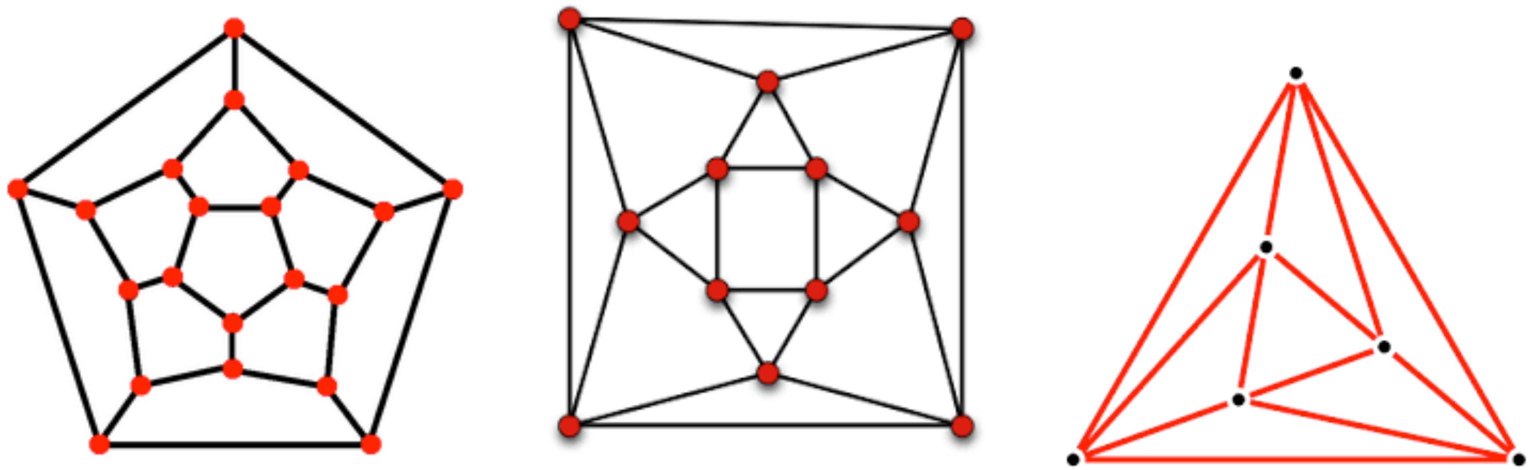


Example 2: Different Intermediate Crossings!



Exercises[a]

1. Why in face routing after choosing the new face you may select either the left or right wall on this face?
2. In the planar graphs below execute face and compass routing between any two pairs of nodes.



3. Indicate the outer face in each of the graphs in Exercise 2.
4. How many faces (including the outer face) do each of the

[a] Do not submit

graphs in Exercise 2 have?

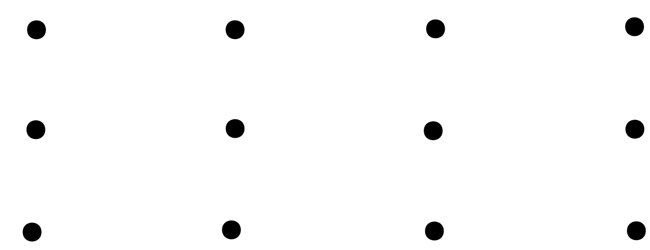
5. There is a formula in graph theory (due to Euler) that says $V + F = E + 2$, where V, E, F is the number of vertices, edges, and faces of a planar graph. Verify that this formula is valid in each of the graphs in Exercise 2.

6. Can you give an example of a planar graph in which face routing from a node to a node t will have to employ the outer face?

7. Consider Rayleigh's principle. Four sensors S_1, S_2, S_3, S_4 which are at distance $1, 2, 3, 4$, respectively, from a sensor S broadcast simultaneously towards S with powers $2, 4, 8, 16$, respectively. Assuming the thrsehold $\lambda = 1$ and the external noise $N = 0$ will S be able to hear the signal of any of the four sensors? If yes, which one?

8. Consider the 12 sensors depicted below. Assume the sensors

have identical range $r > 0$



and horzontal distances are 2 units while vertical are 1 unit.

(a) Draw the resulting UDG when $r < 1$.

(b) Draw the resulting UDG when $r = 1$.

(c) Draw the resulting UDG when $r = 1.5$.

(d) Draw the resulting UDG when $r = 2$.

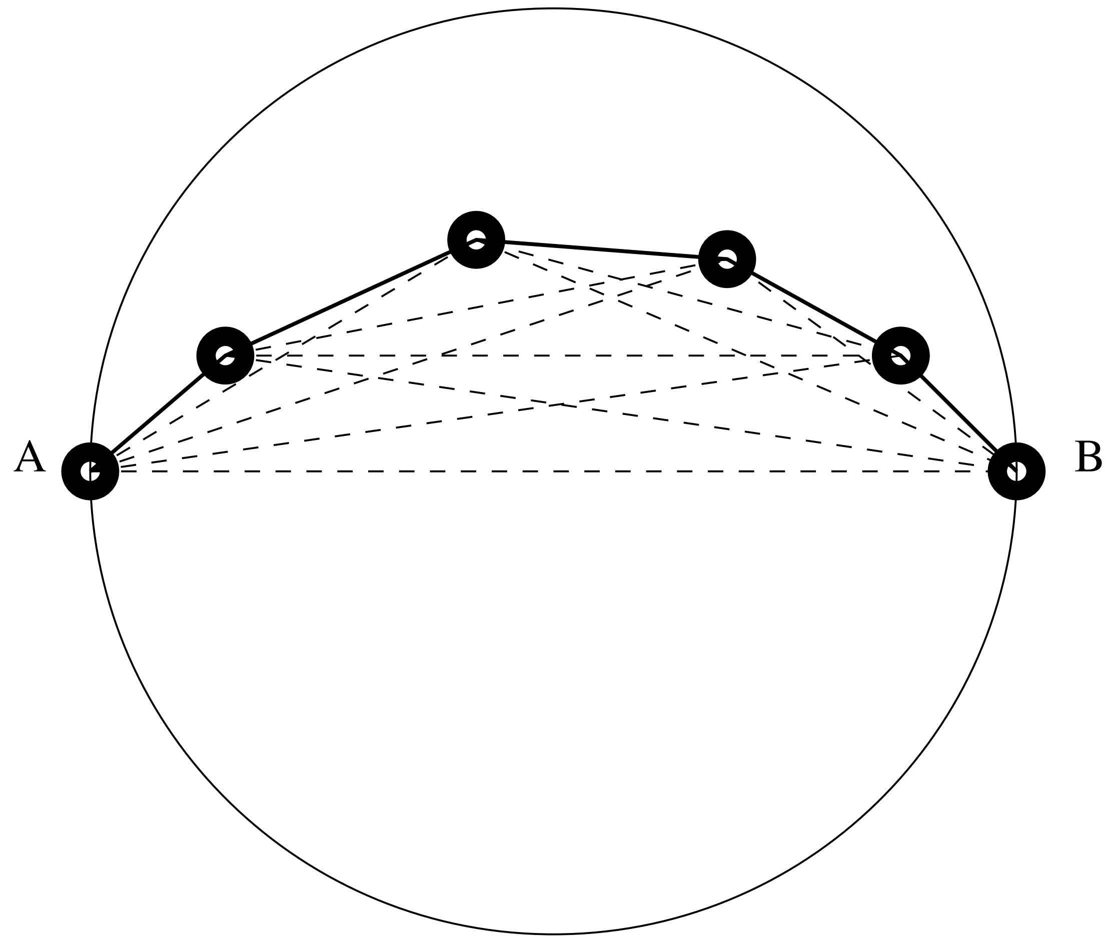
(e) Draw the resulting UDG when $r = \sqrt{5}$.

(f) Draw the resulting UDG when $r = 4$.

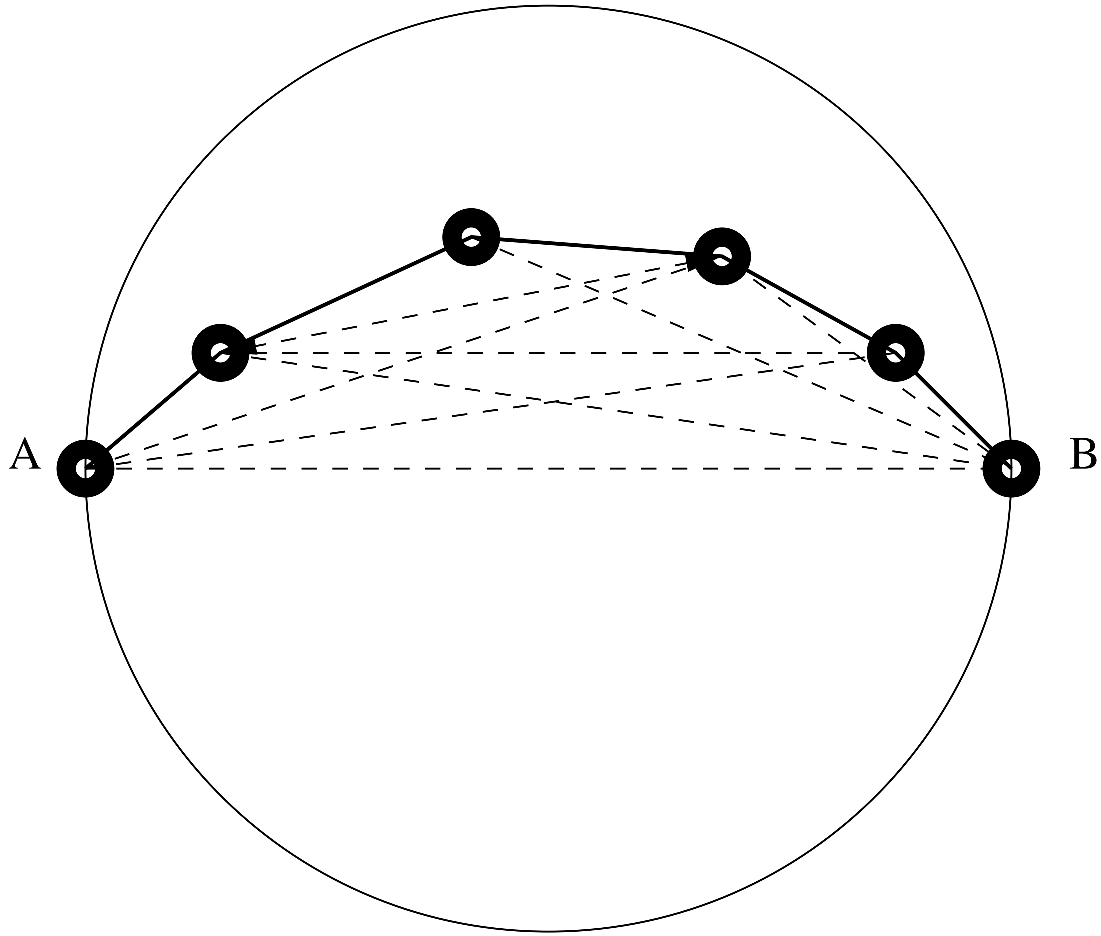
9. Apply the Gabriel Test in any of the UDGs of Exercise 8 and draw the resulting graph.

Appendix

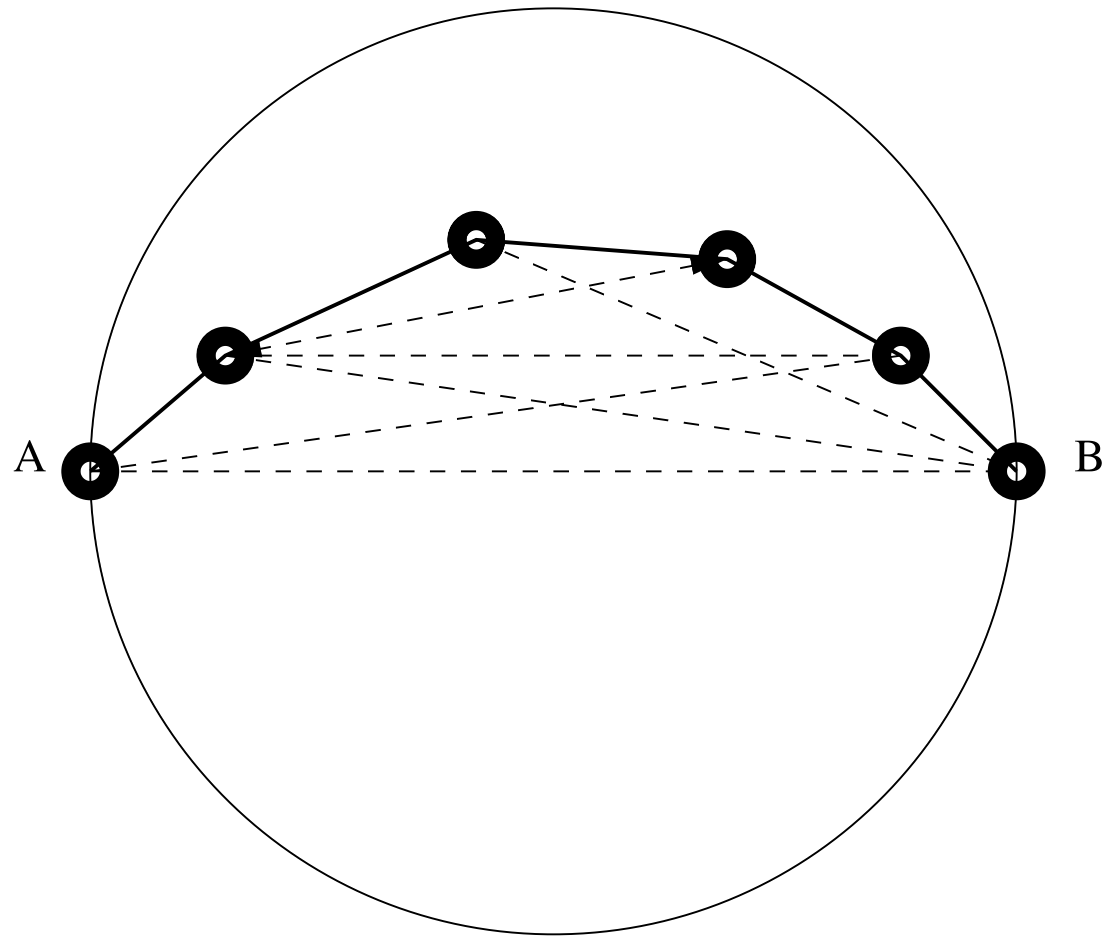
Applying the Gabriel Test (1/9)



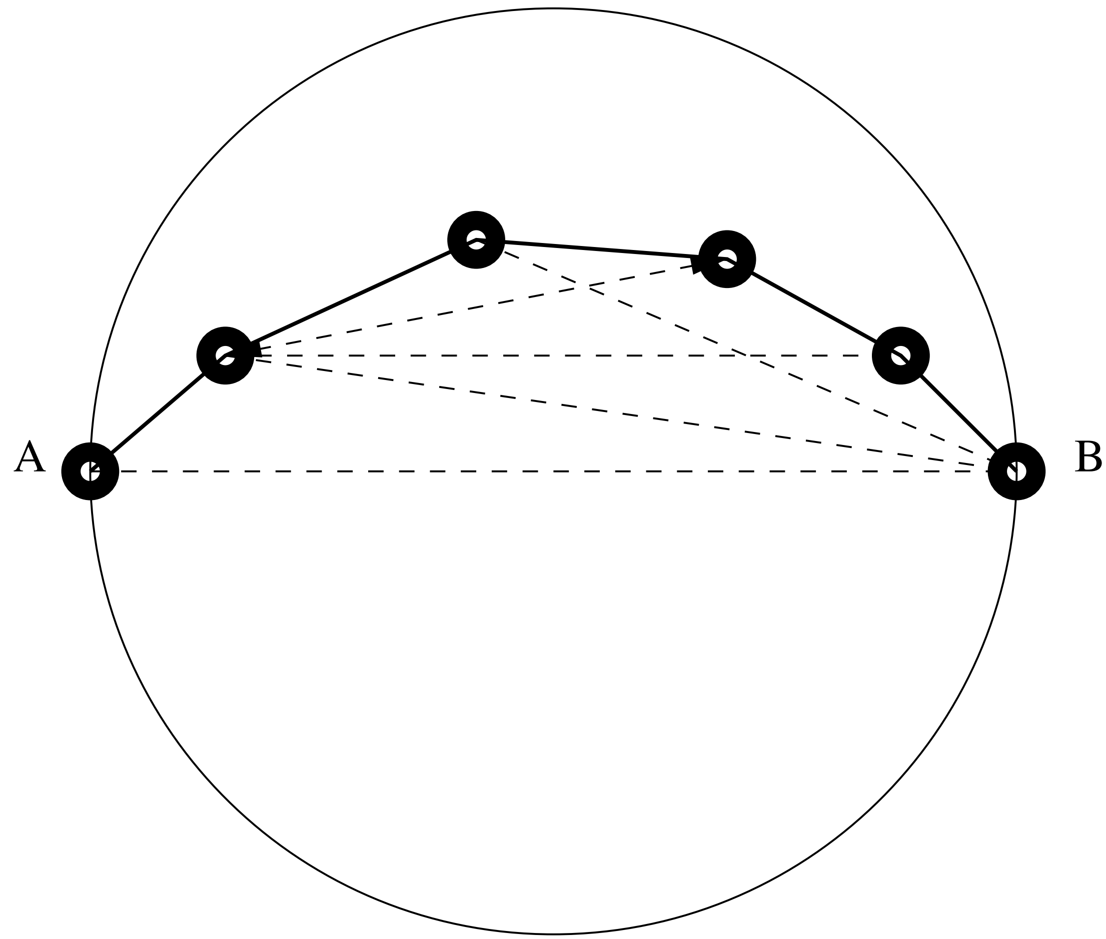
Applying the Gabriel Test (2/9)



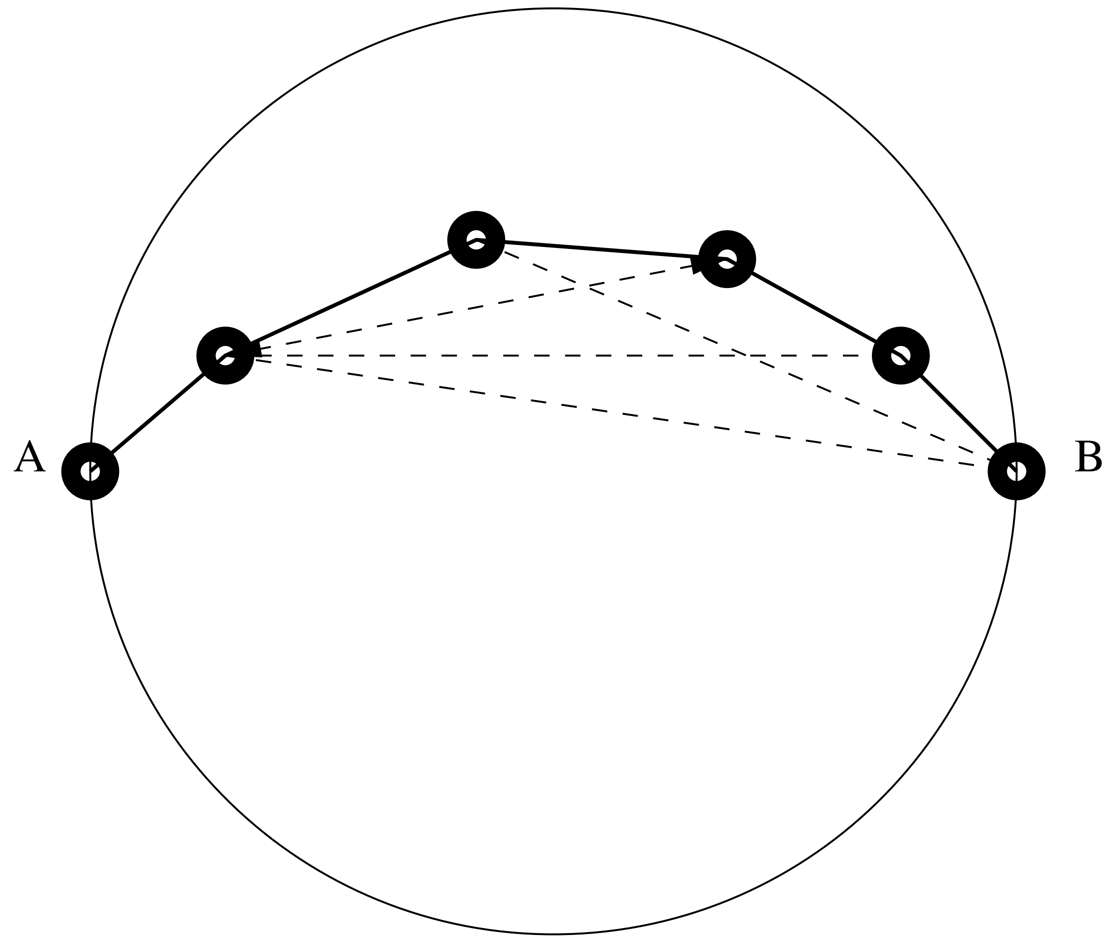
Applying the Gabriel Test (3/9)



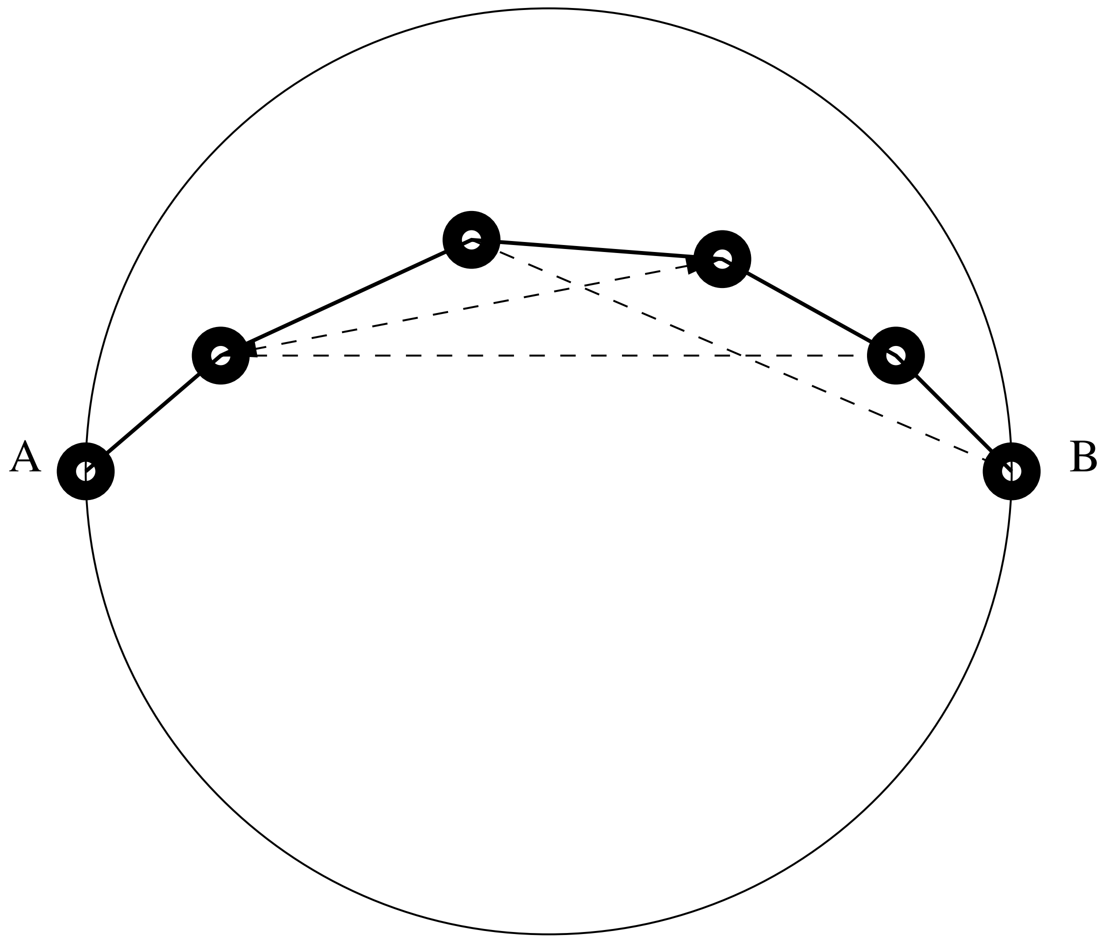
Applying the Gabriel Test (4/9)



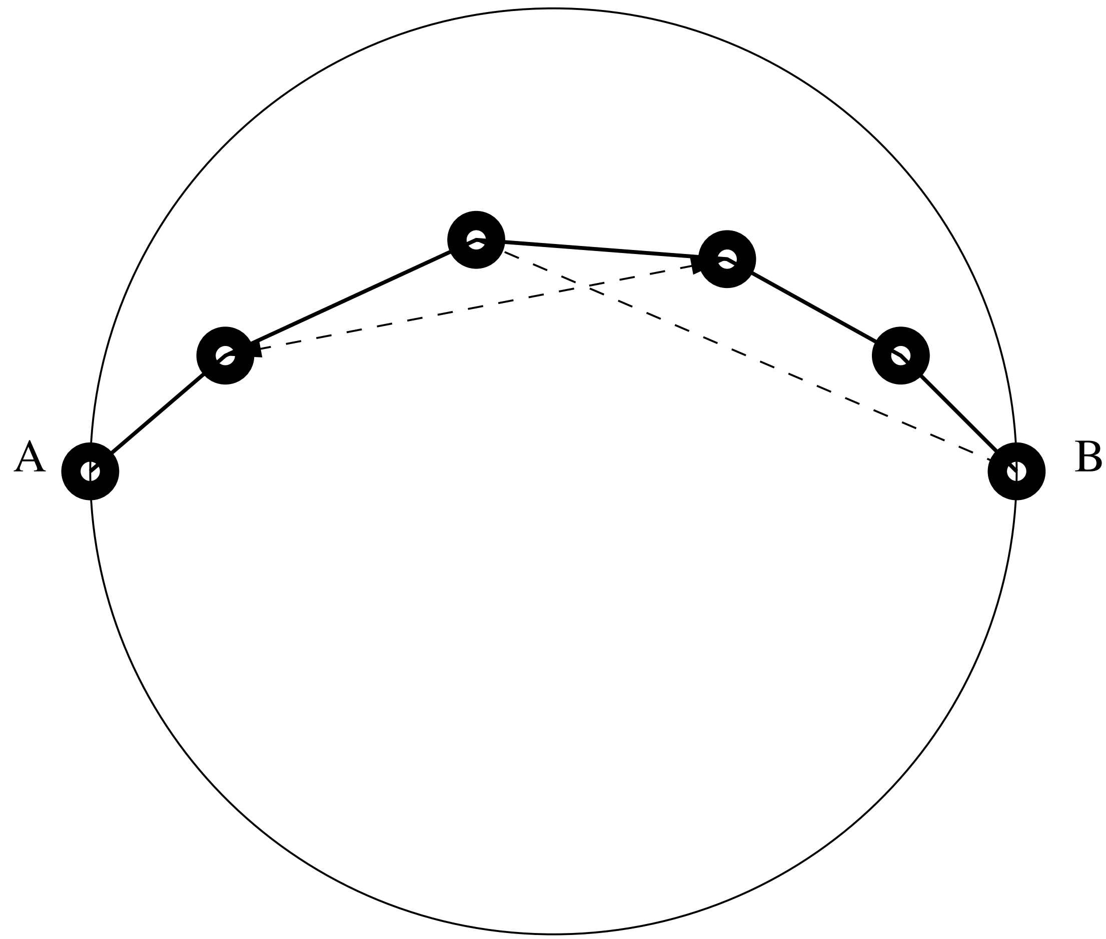
Applying the Gabriel Test (5/9)



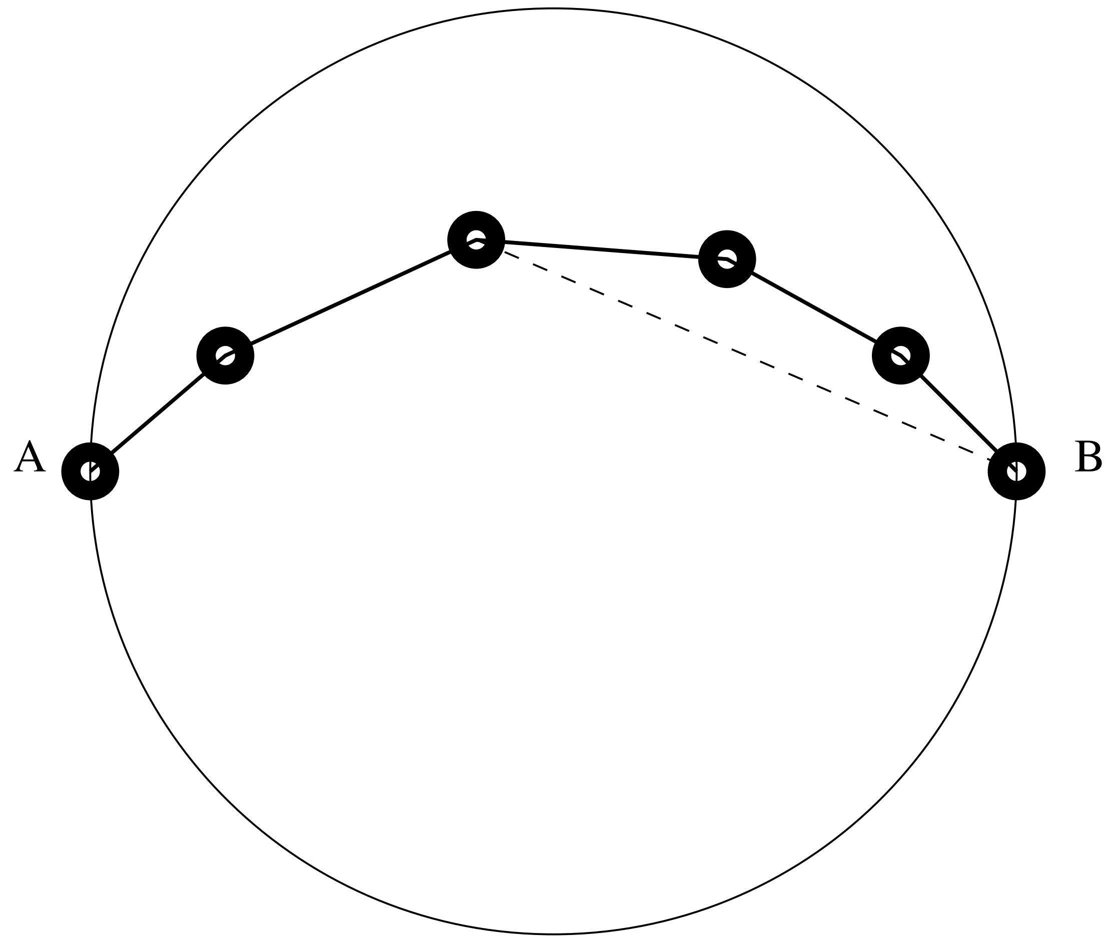
Applying the Gabriel Test (6/9)



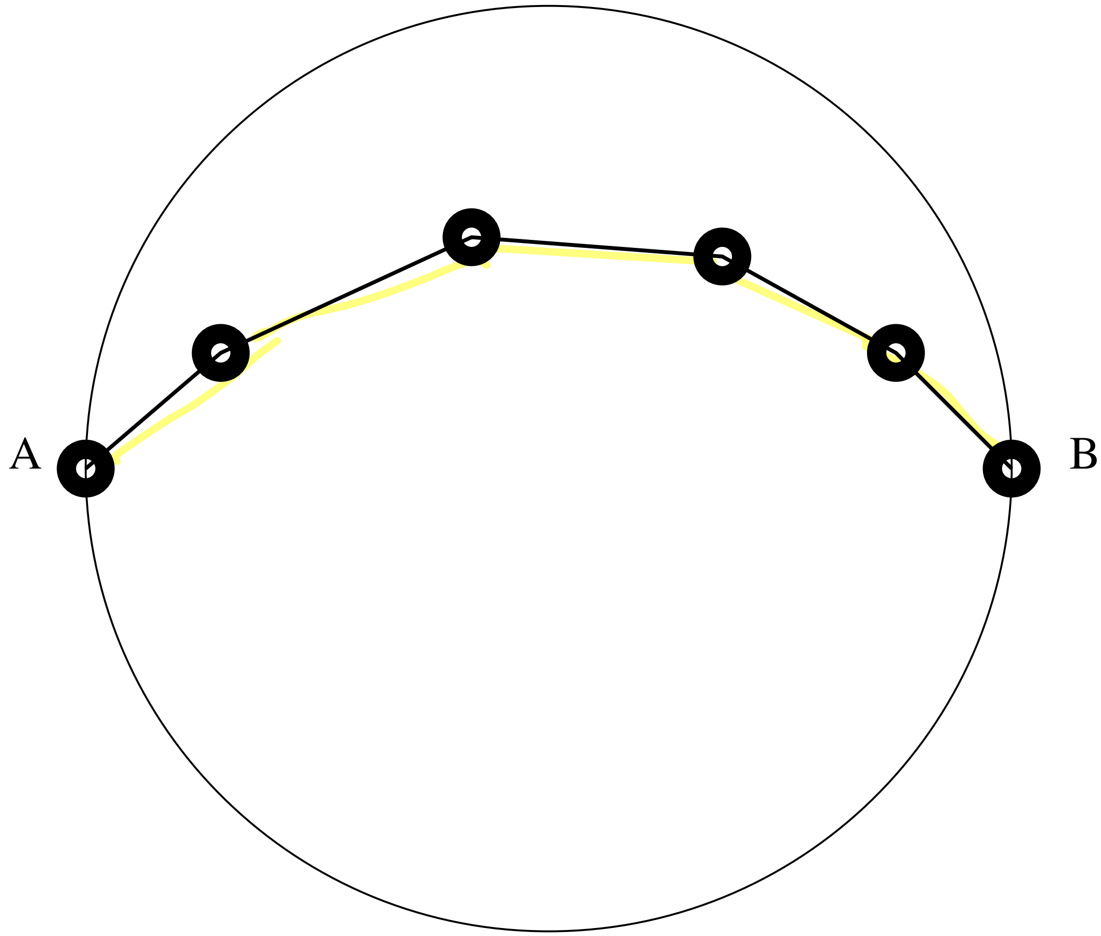
Applying the Gabriel Test (7/9)



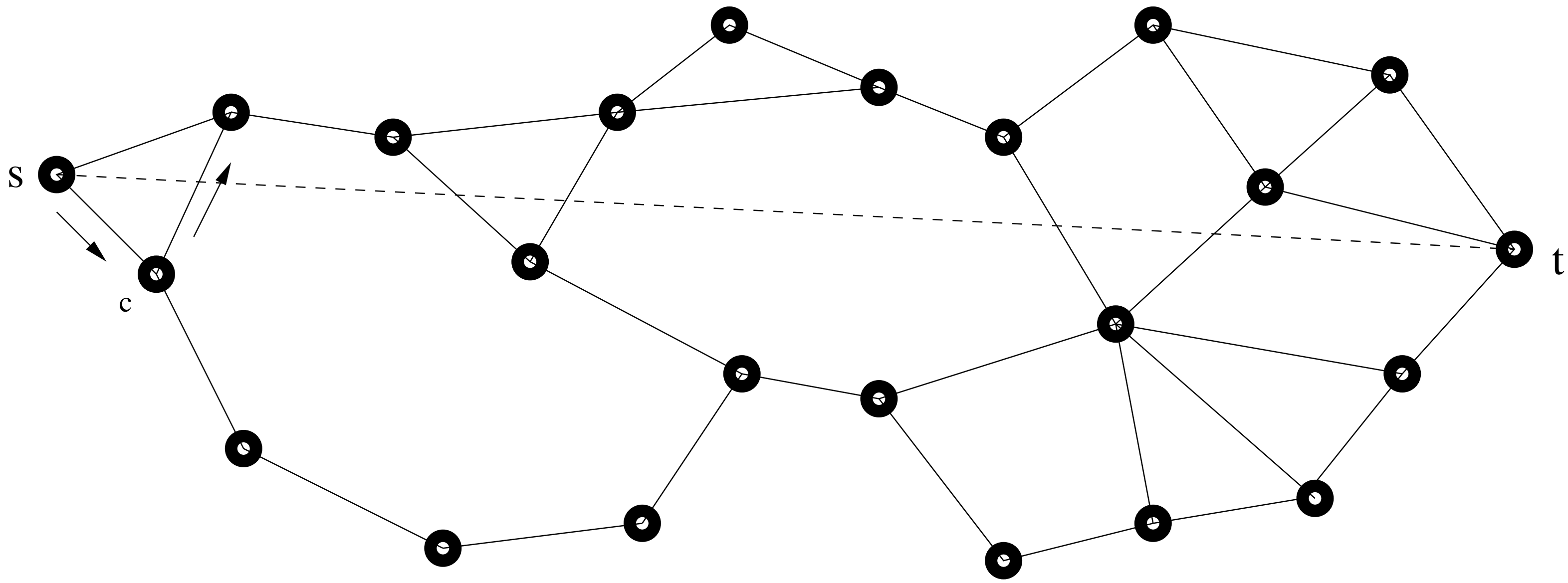
Applying the Gabriel Test (8/9)



Applying the Gabriel Test (9/)9



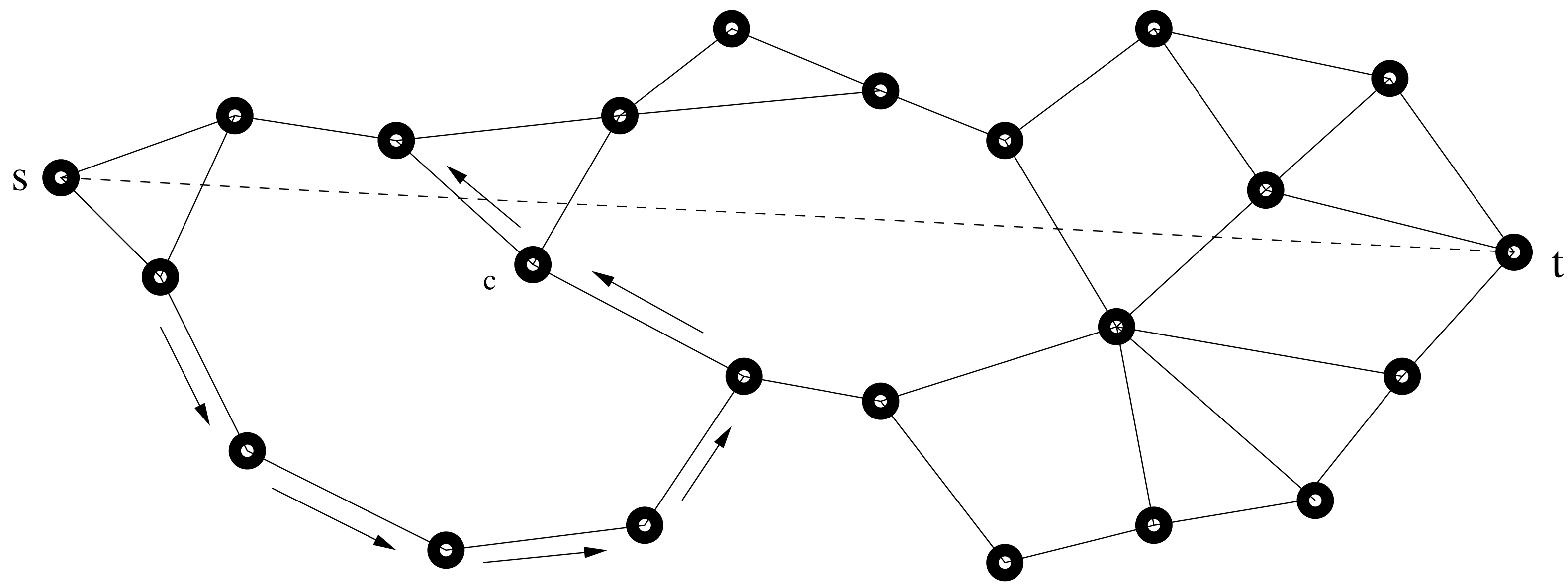
Example: Go from s to t



Initially $c := s$.

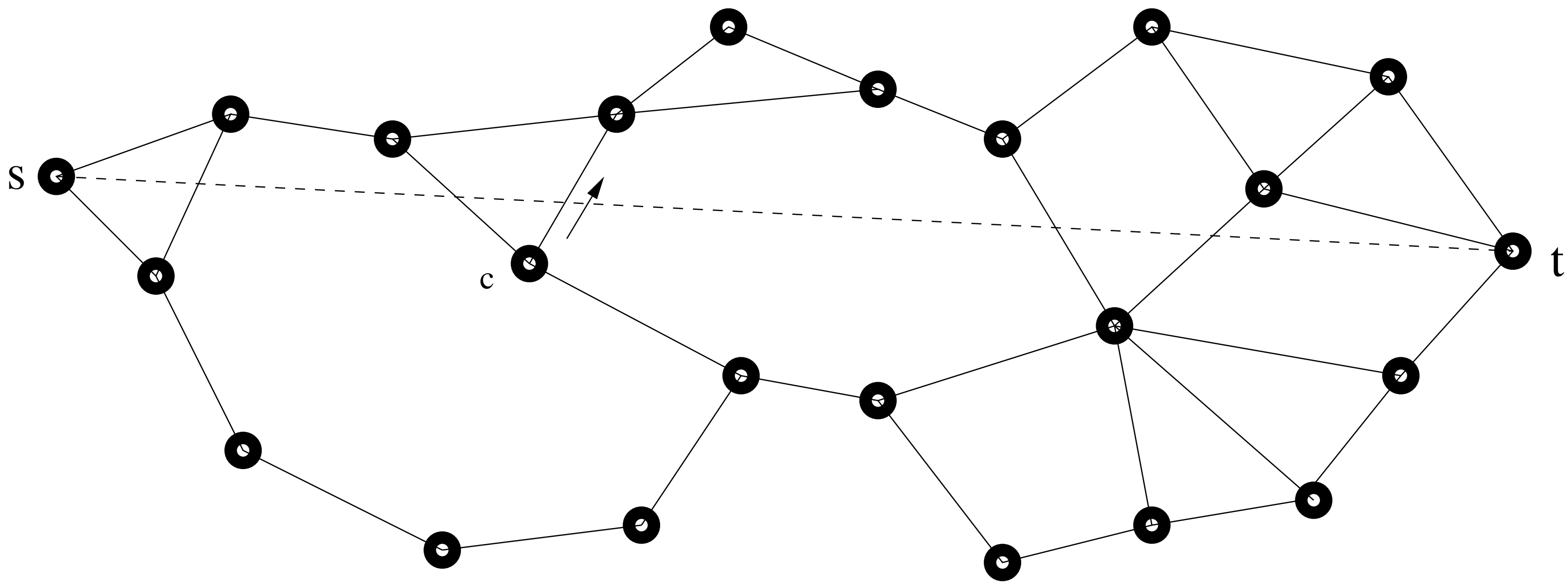
Update c and repeat.

Go from s to t



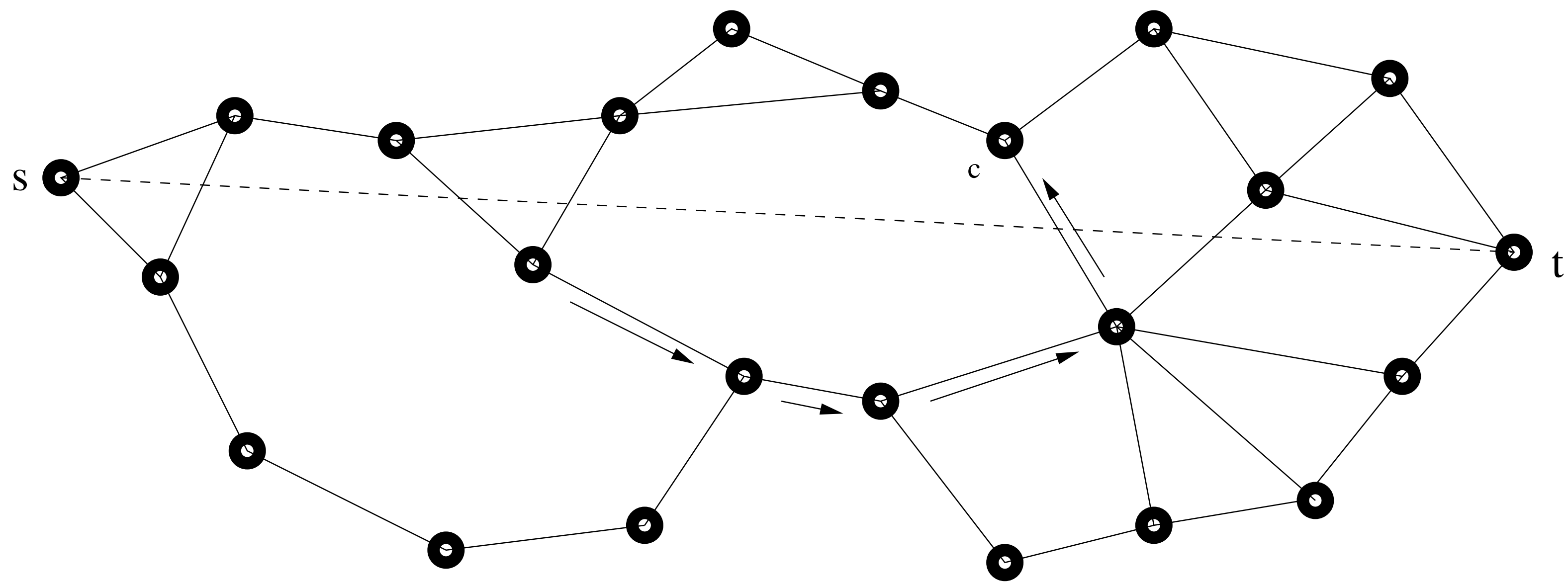
Update c and repeat.

Go from s to t



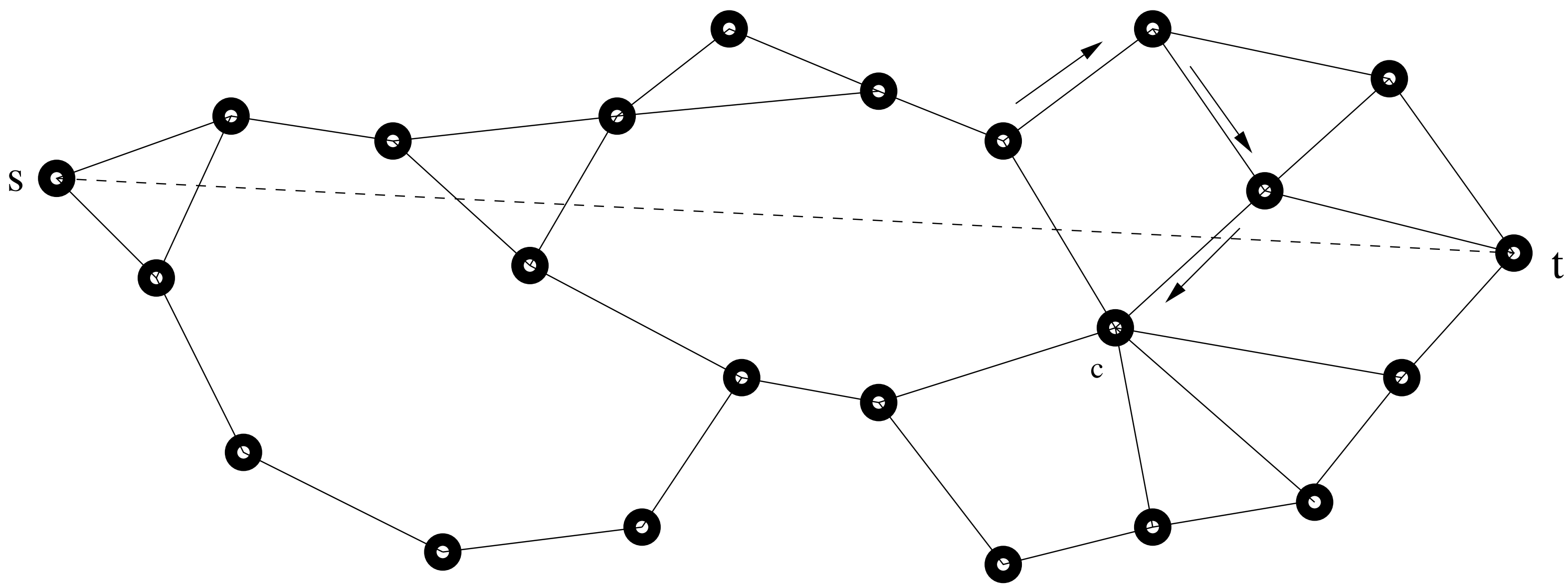
Update c and repeat.

Go from s to t



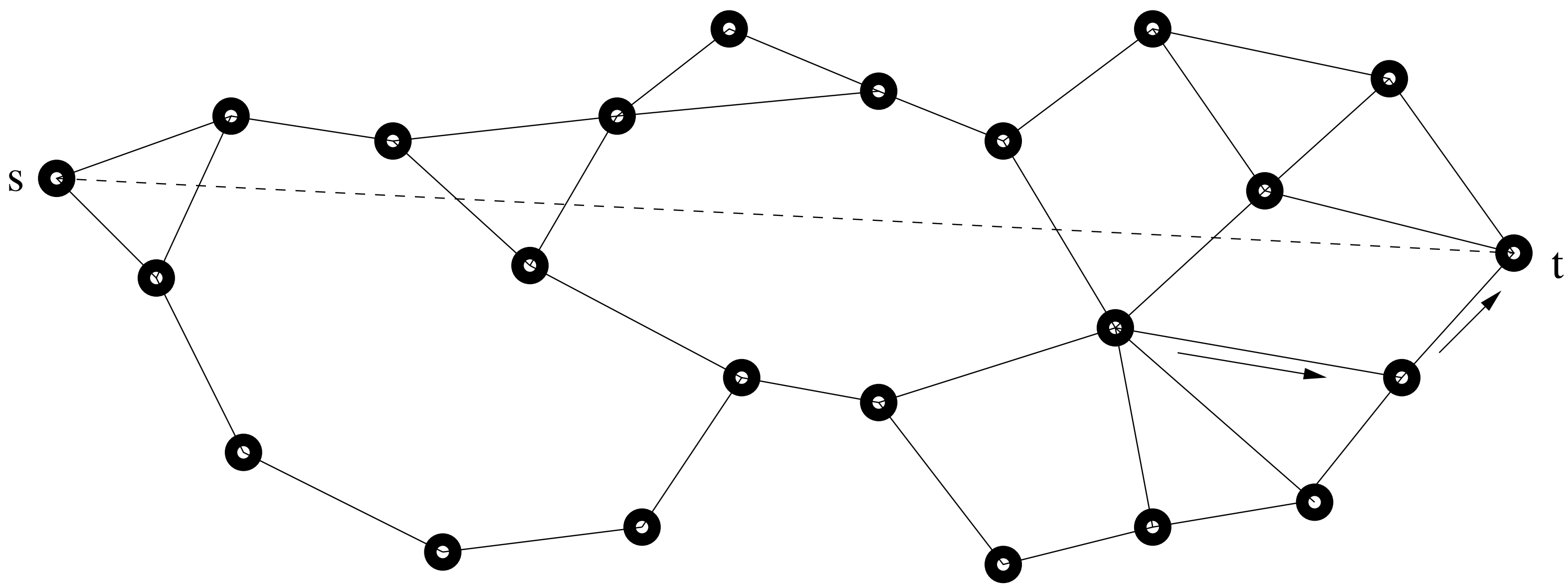
Update c : and repeat.

Go from s to t



Update c : and repeat.

Go from s to t



Now t is found.