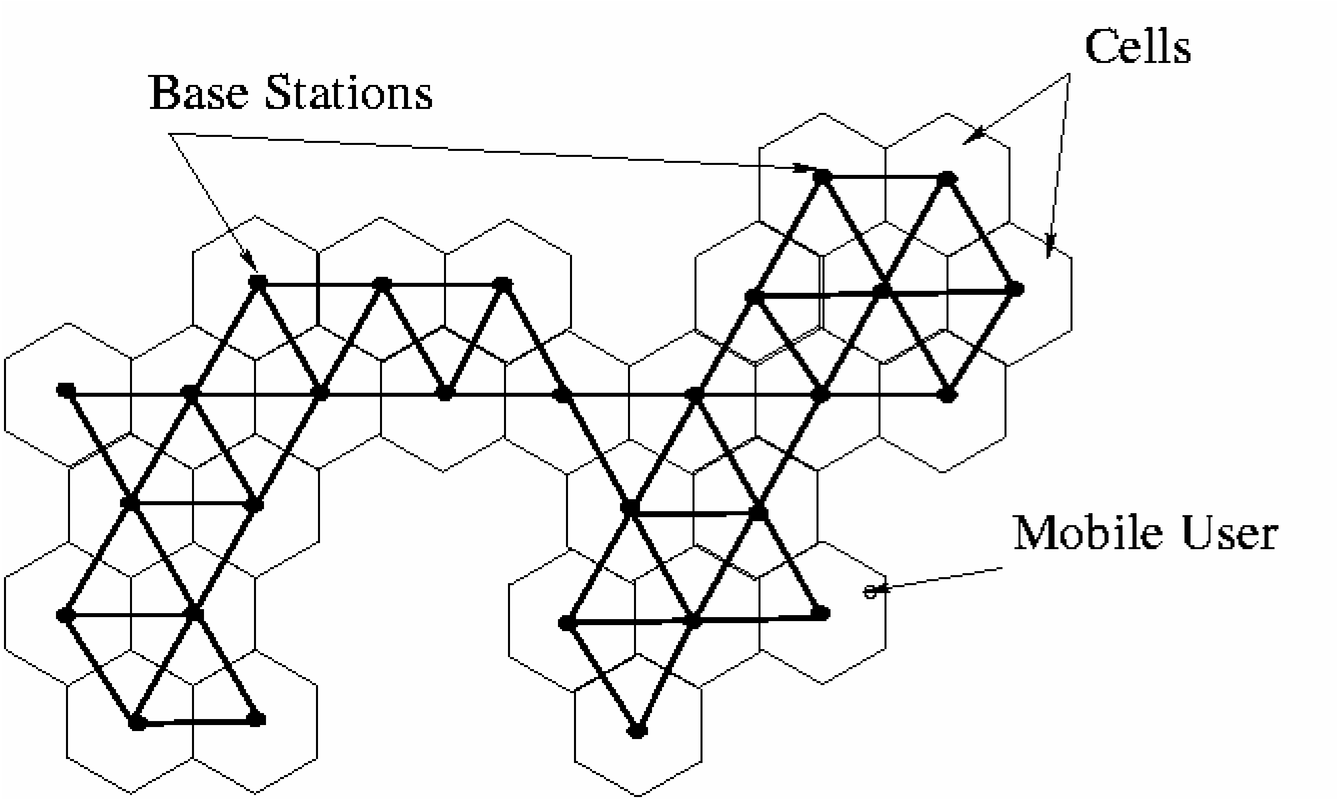
Addressing, Distances and Routing in Triangular Systems with Application in Cellular and Sensor Networks

Victor Chepoi, Feodor Dragan, Yan Vaxes

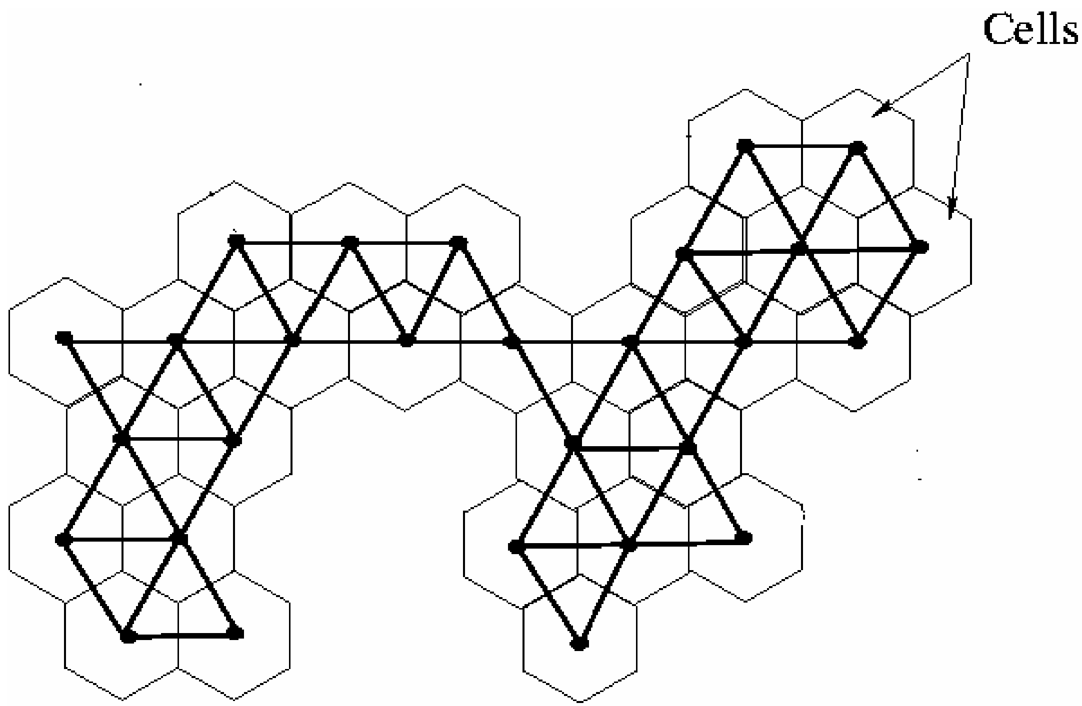
University of Marseille, France
Kent State University, Ohio, USA

Cellular Network



Benzenoid and Triangular Systems

- Benzenoid Systems: is a simple circuit of the hexagonal grid and the region bounded by this circuit.



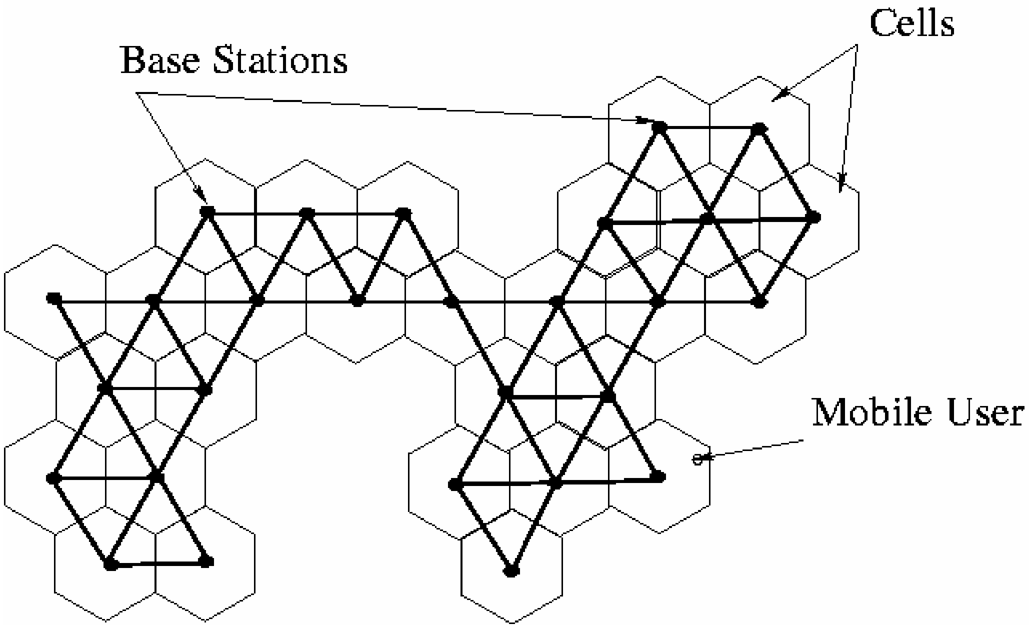
- The Duals to Benzenoid Systems are Triangular Systems

Addressing, Distances and Routing in Triangular Systems: Motivation

Applications in cellular networks

- Identification code (CIC) for tracking mobile users
- Dynamic location update (or registration) scheme
  - time based
  - movement based
  - distance based

(cell-distance based is best, according to [Bar-Noy&Kessler&Sidi'94])



- ➔Distances
- Routing protocol

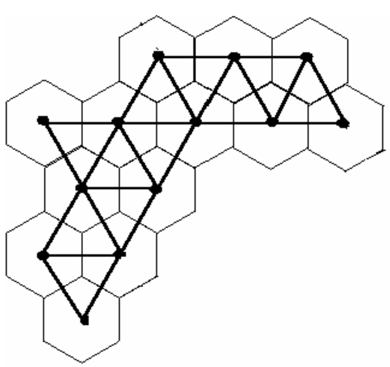
Current situation

- Current cellular networks do not provide information that can be used to derive cell distances
  - It is hard to compute the distances between cells (claim from [Bar-Noy&Kessler&Sidi'94])
  - It requires a lot of storage to maintain the distance information among cells (claim from [Akyildiz&Ho&Lin'96] and [Li&Kameda&Li'00])

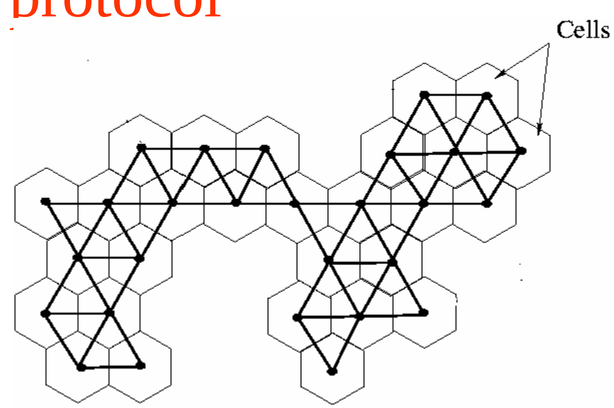
Recent results

[Nocetti&Stojmenovic&Zhang'02] recently considered isometric subgraphs of the regular triangular grid and give among others

- A new cell addressing scheme using only three small integers, one of them being zero
- A very simple method to compute the distance between two sells
- A short and elegant routing protocol



isometric

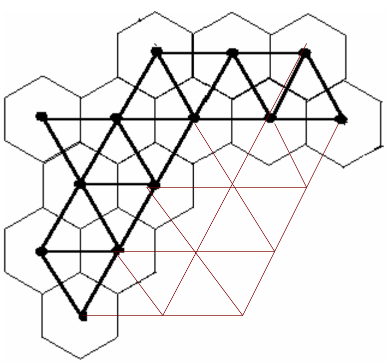


not isometric

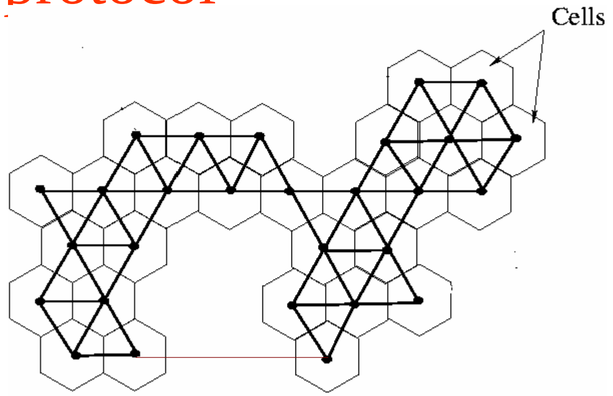
Recent results

[Nocetti&Stojmenovic&Zhang'02] recently considered isometric subgraphs of the regular triangular grid and give among others

- A new cell addressing scheme using only three small integers, one of them being zero
- A very simple method to compute the distance between two sells
- A short and elegant routing protocol



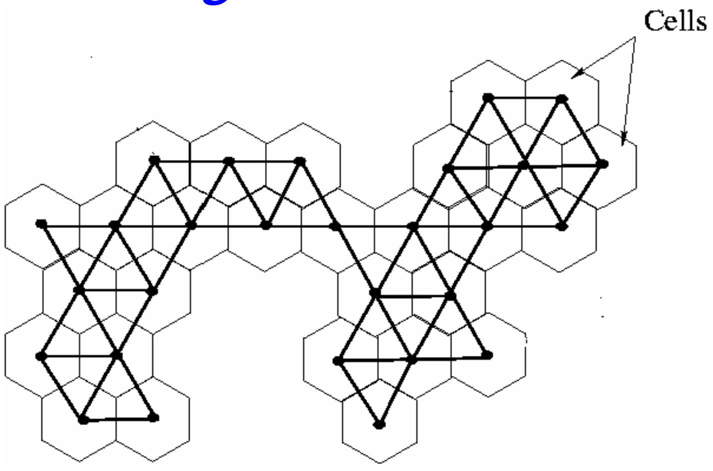
isometric



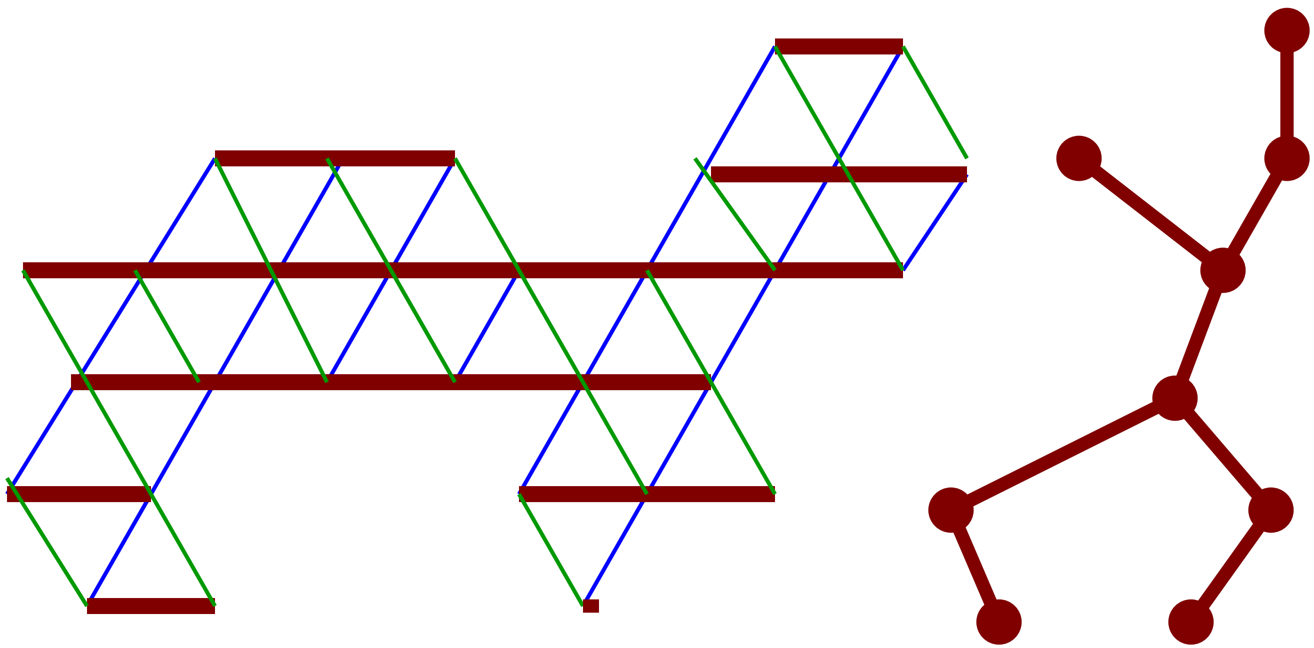
not isometric

Our results for triangular systems

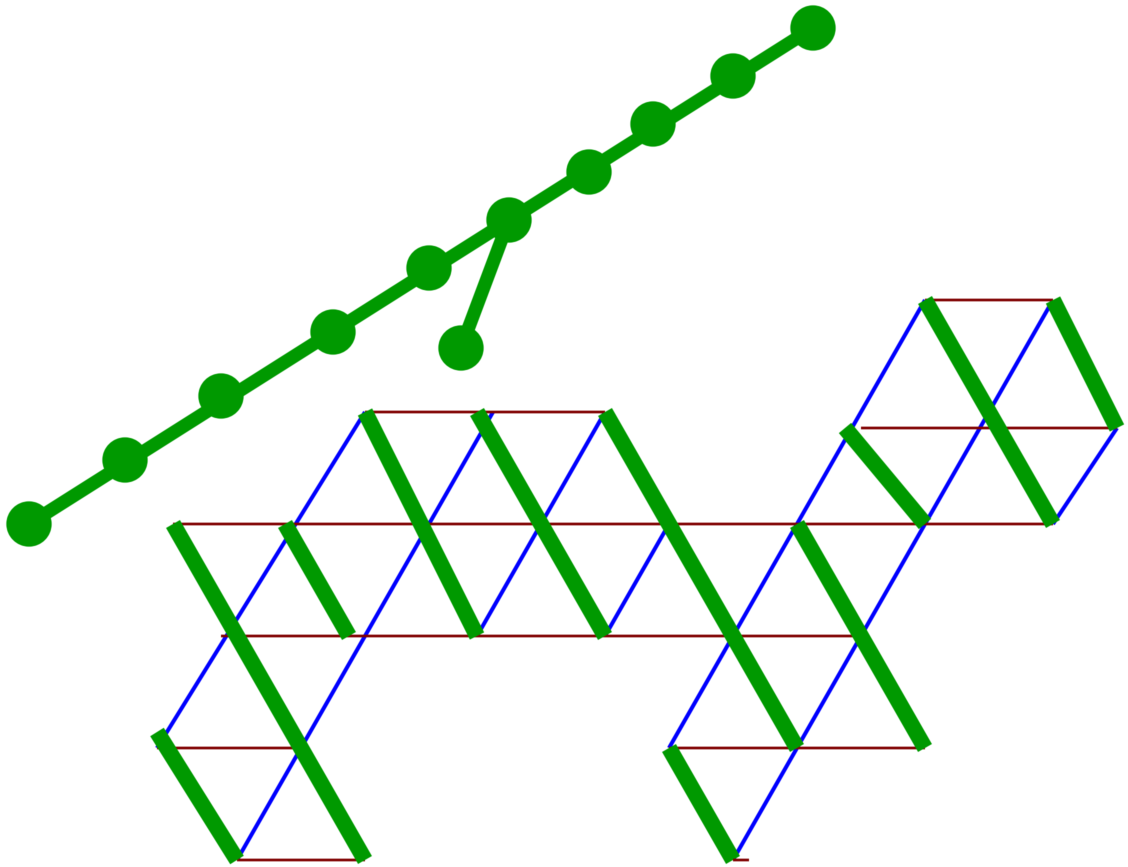
- Scale 2 isometric embedding into Cartesian product of 3 trees
  - ➔ cell addressing scheme using only three small integers
  - ➔ distance labeling scheme with labels of size $O(\log^2 n)$ -*bits per node and constant time distance decoder*
  - ➔ routing labeling scheme with labels of size $O(\log n)$*-bits per node and constant time routing decision.*



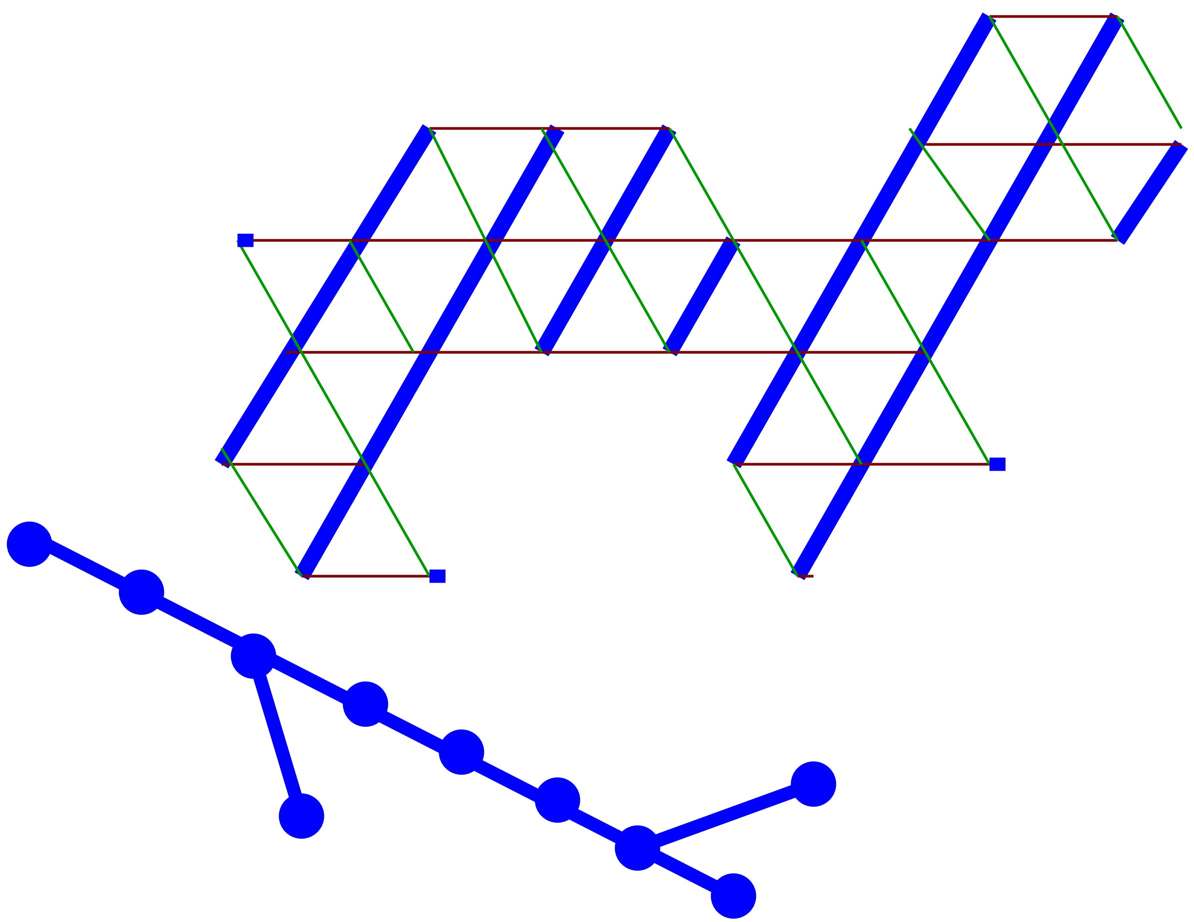
Three edge directions➔three trees



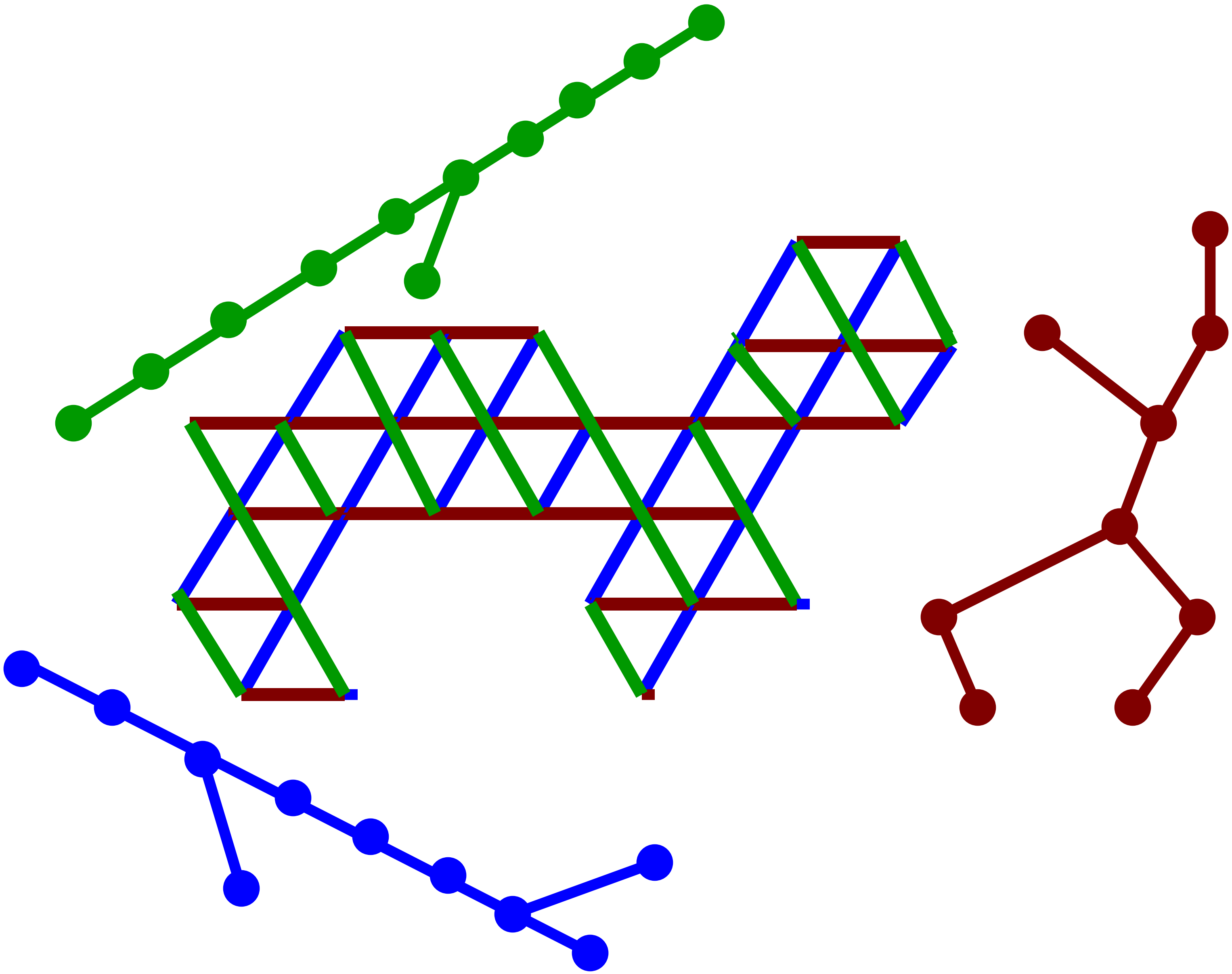
Three edge directions→three trees



Three edge directions→three trees

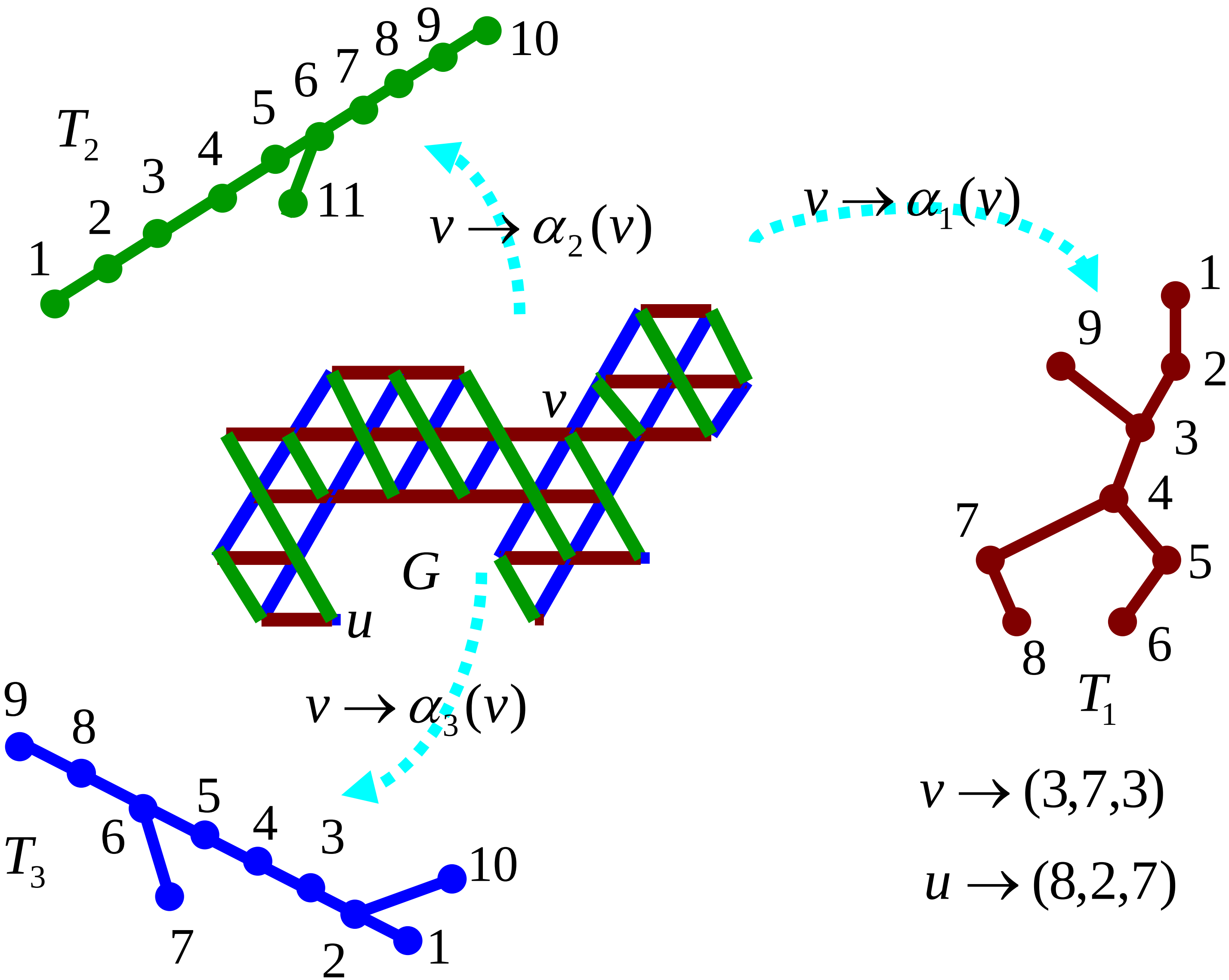


Three edge directions➔three trees



Addressing

$$v \to (\alpha_1(v), \alpha_2(v), \alpha_3(v))$$



$$v \to (3,7,3)$$

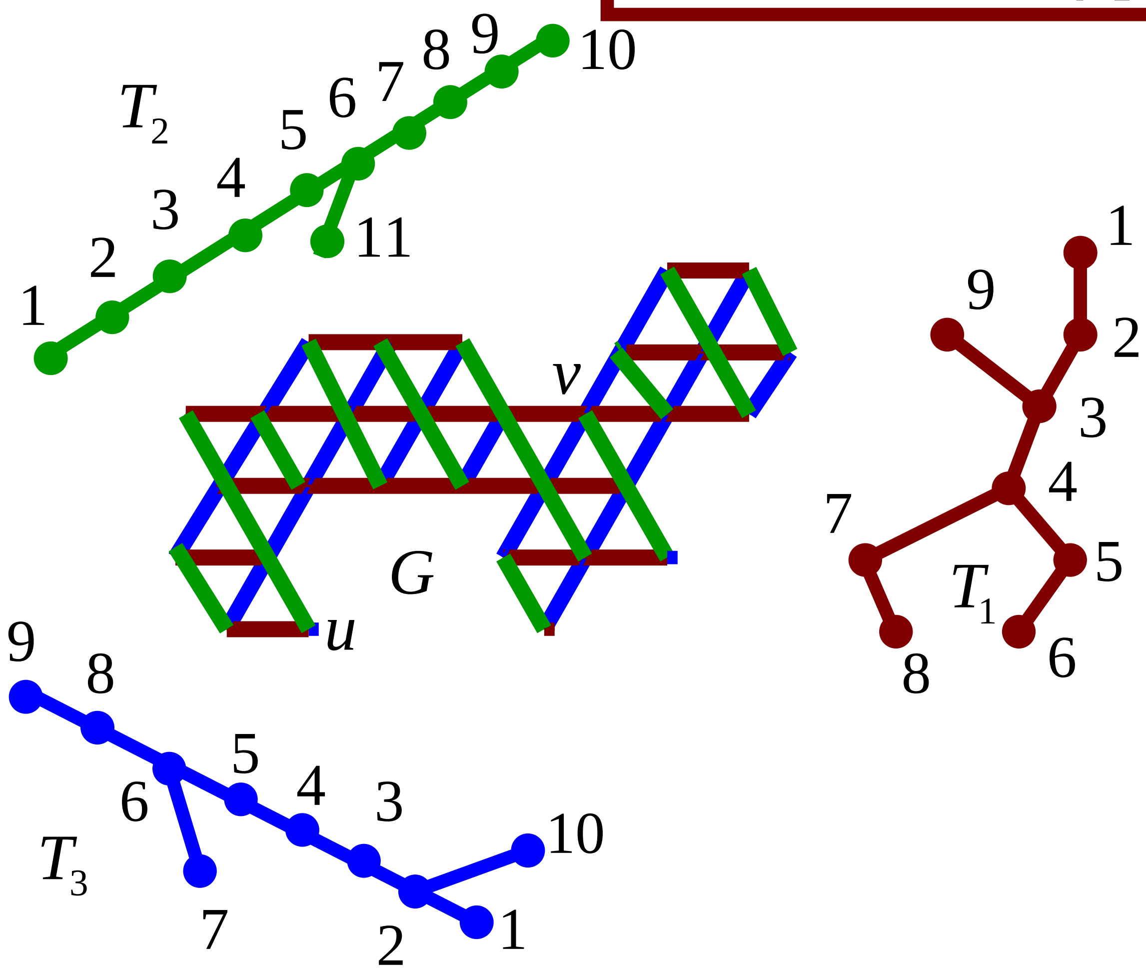
$$u \to (8,2,7)$$

Scale 2 embedding into 3 trees

$$v \to (\alpha_1(v), \alpha_2(v), \alpha_3(v))$$

$$u \to (\alpha_1(u), \alpha_2(u), \alpha_3(u))$$

$$2dist_G(v,u) = \sum_{i=1}^{3} dist_{T_i}(\alpha_i(v), \alpha_i(u))$$



$$v \to (3,7,3)$$

$$u \to (8,2,7)$$

$$dist_{T_1}(\alpha_1(v), \alpha_1(u)) +$$
$$dist_{T_2}(\alpha_2(v), \alpha_2(u)) +$$
$$dist_{T_3}(\alpha_3(v), \alpha_3(u)) =$$
$$3 + 5 + 4 = 12 =$$
$$2dist_G(v,u)$$

Distance Labeling Scheme

Goal: Short labels that encode distances and distance decoder, an algorithm for inferring the distance between two nodes only from their labels (in time polynomial in the label length)

- **Labeling:** *v* → *Label(v)*

  (for trees, $O(\log^2 n)$ *bits per node* [Peleg'99]*)*

- **Distance decoder:** *D(Label(v), Label(u))* → *dist(u,v)*

  (for trees, constant decision time)

Distance labeling scheme for triangular systems

- Given *G*, find three corresponding trees T_1, T_2, T_3 and addressing $v \rightarrow (\alpha_1(v), \alpha_2(v), \alpha_3(v))$ (*O(n) time*)

- Construct distance labeling scheme for each tree

  $\alpha_i(v) \rightarrow Label(\alpha_i(v))$ (*O(n*log*n) time*)

- Then, set $Label(v) = (Label(\alpha_1(v)), Label(\alpha_2(v)), Label(\alpha_3(v)))$

- $O(\log^2 n)$ -bit labels and constructible in total time $O(n \log n)$

Distance decoder for triangular systems

Given *Label(u)* and *Label(v)*

Function distance_decoder_triang_syst(Label(u),Label(v))

- Output ***½(distance_decoder_trees(*** $Label(\alpha_1(v)), Label(\alpha_1(u))$ ***)***

  +(distance_decoder_trees($Label(\alpha_2(v)), Label(\alpha_2(u))$ ***)***

  +(distance_decoder_trees($Label(\alpha_3(v)), Label(\alpha_3(u))$ ***))***

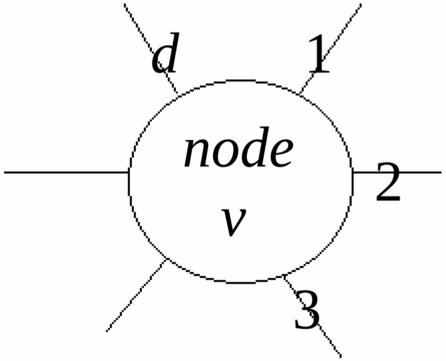
Thm: The family of *n*-node triangular systems enjoys a distance labeling scheme with $O(\log^2 n)$ -bit labels and a constant time distance decoder.

Routing Labeling Scheme

Goal: Short labels that encode the routing information and routing protocol, an algorithm for inferring port number of the first edge on a shortest path from source to destination, giving only labels and nothing else

- **Labeling:** *v* → *Label(v)*
- **Distance decoder:** *R(Label(v), Label(u))* → *port(v,u)*

(for trees, $O(\log n)$ *bits per node and constant time decision* [Thorup&Zwick'01]*)*



Routing labeling scheme for triangular systems

- Given *G*, find three corresponding trees T_1, T_2, T_3 and addressing $v \to (\alpha_1(v), \alpha_2(v), \alpha_3(v))$

- Construct routing labeling scheme for each tree using Thorup&Zwick method (*log n* bit labels)

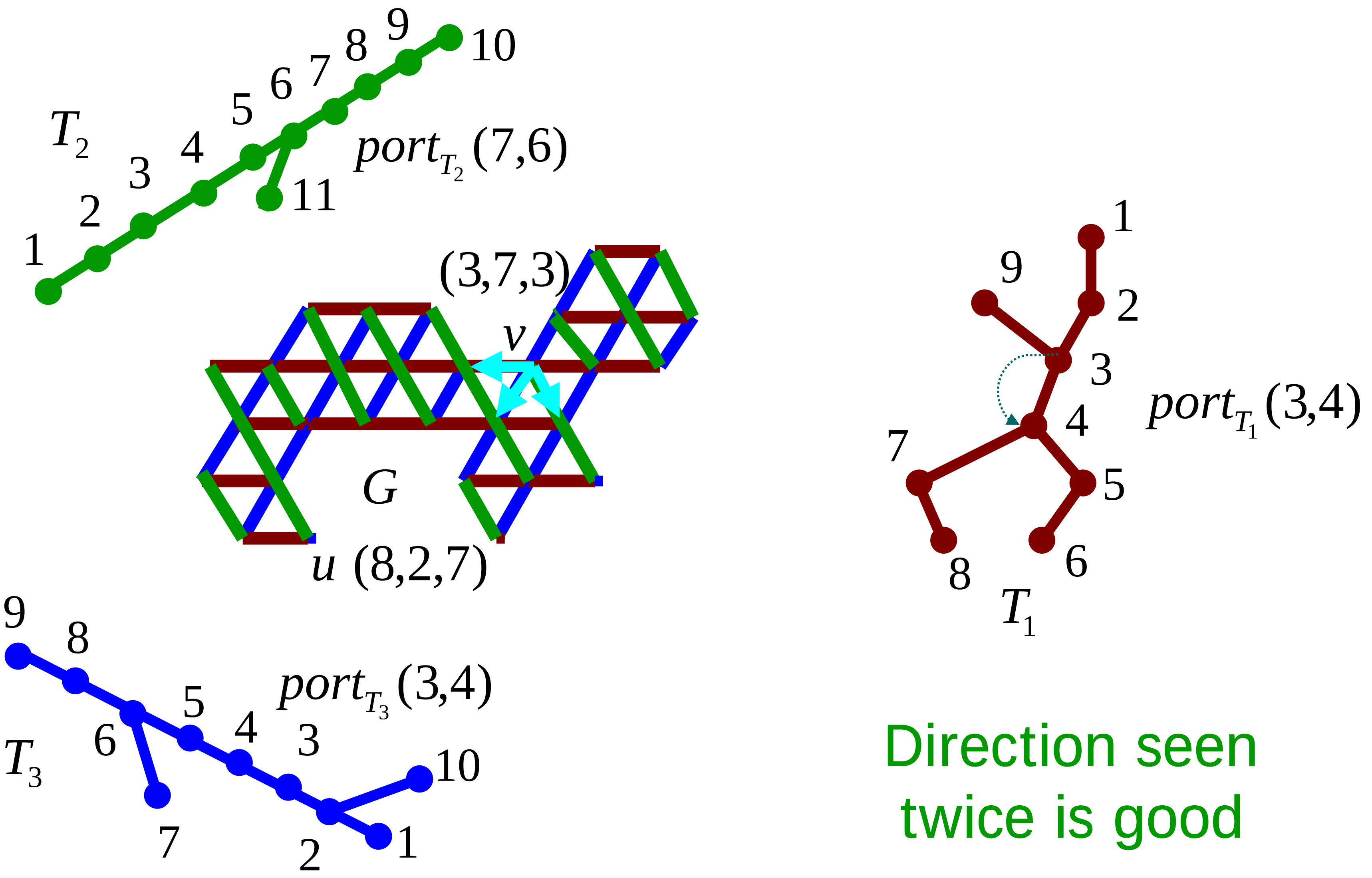
$$\alpha_i(v) \to Label(\alpha_i(v))$$

- Then, set

$$Label(v) = (Label(\alpha_1(v)), Label(\alpha_2(v)), Label(\alpha_3(v)), \ldots.)$$

Something more

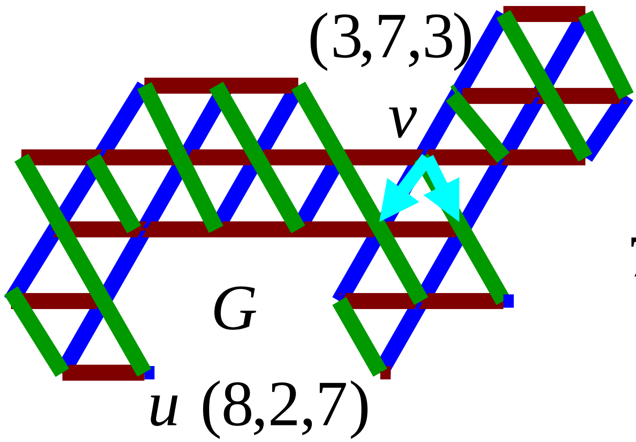
Choosing direction to go from v

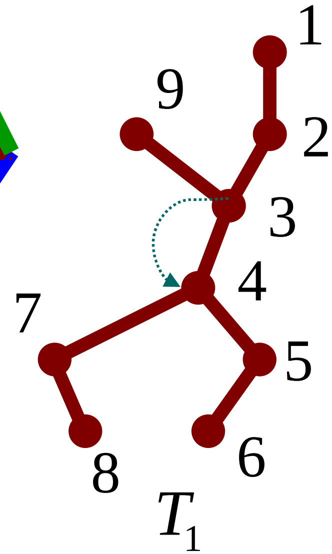


Direction seen
twice is good

Mapping tree ports to graph ports

$$\Delta_i(v) = (\alpha_i^1(v),\ldots,\alpha_i^j(v)) = (2,4) \quad (j \le 4)$$





$$O_i(v) = ((port_{T_i}(\alpha_i(v),\alpha_i^1(v)), Q_i^1(v)),\ldots,(port_{T_i}(\alpha_i(v),\alpha_i^j(v)), Q_i^j(v)) =$$

$$((port_{T_1}(3,2), Q_1^1(v)),\ldots,(port_{T_1}(3,4), Q_1^2(v))$$

$$Q_i^j(v) = (port_G^1, port_G^2)$$

Then, $Label(v) = (Label(\alpha_1(v)), Label(\alpha_2(v)), Label(\alpha_3(v)),\ldots.)$

(i.e., *3xlog n+3x4x3xlogn* bit labels)

Routing Decision for triangular systems

Given *Label(u)* and *Label(v)*

```
function routing_decision_triang_syst(L(x), H(y))

if (α_1(x), α_2(x), α_3(x)) = (α_1(y), α_2(y), α_3(y)) then return "packet reached its destination";
set A ← 0;
for each i ∈ {1,2,3} do
   p ← routing_decision_trees(L_{T_i}(α_i(x)), H_{T_i}(α_i(y)));
   for each j ∈ {1,...,|O_i(x)|} do
      if p = O_i(x)[j] then
         for each entry port_G of the array Q_i^j(x) do
            A[port_G] ← A[port_G] + 1;
            if A[port_G] = 2 then
               return port_G.
```

Thm: The family of n-node triangular systems enjoys a routing labeling scheme with $O(\log n)$-bit labels and a constant time routing decision.

We showed for triangular systems

- Scale 2 isometric embedding into Cartesian product of 3 trees
  - ➔ cell addressing scheme using only three small integers
  - ➔ cell-distance labeling scheme with labels of size $O(\log^2 n)$-*bits per node and constant time distance decoder*
  - ➔ routing labeling scheme with labels of size $O(\log n)$-*bits per node and constant time routing decision.*

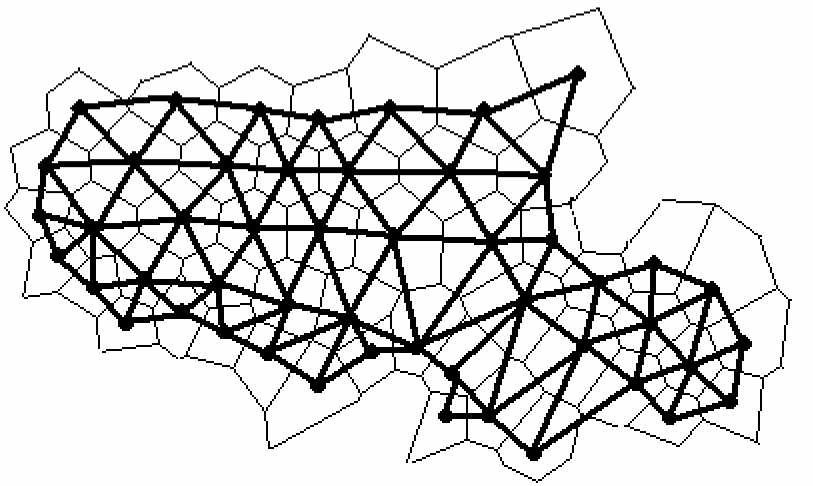
Extensions, Open Problems

- BS at arbitrary positions

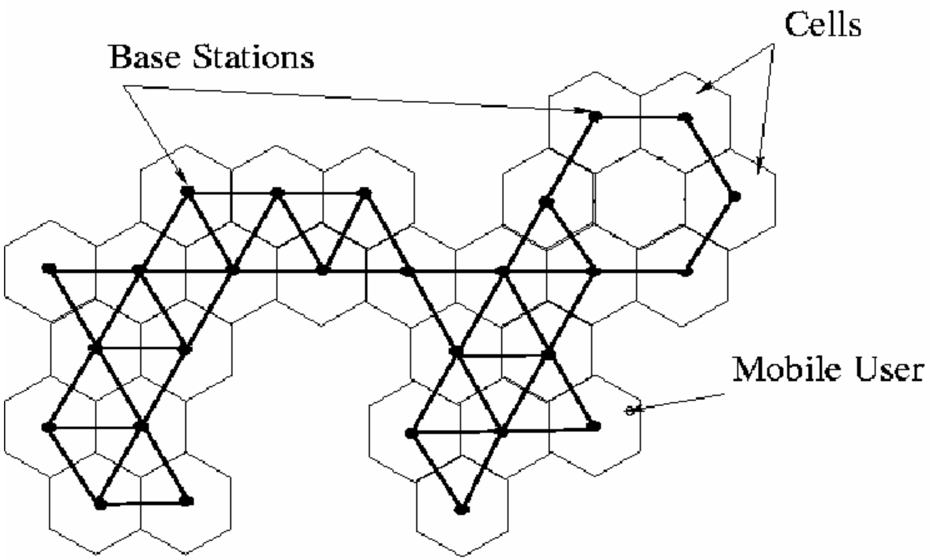
➔ BS determine

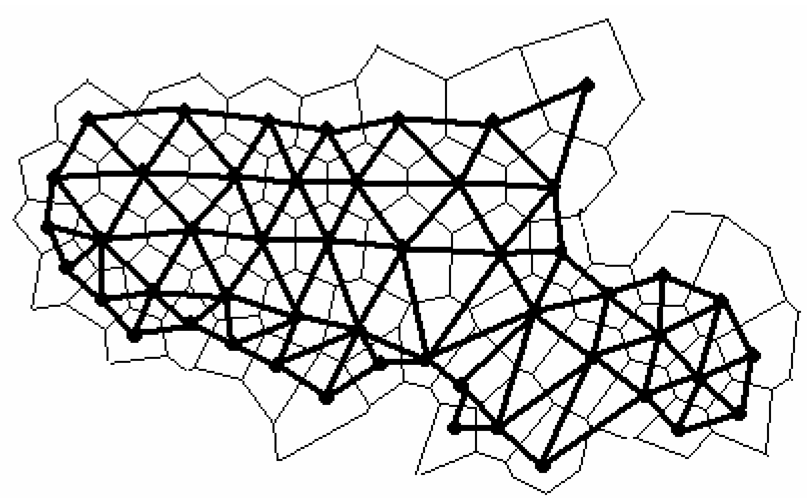
Delaunay triangulation

= dual Voronoi diagram



- Not-Simply Connected Cellular Networks



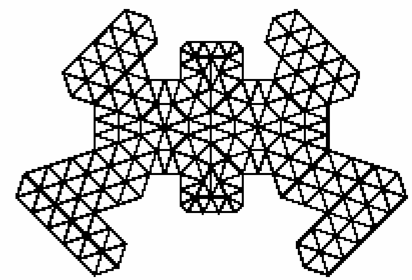


Other Results

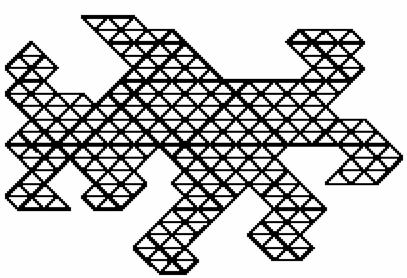
Thm: The families of n-node (6,3)-,(4,4)-,(3,6)-planar graphs enjoy distance and routing labeling schemes with $O(\log^2 n)$ -bit labels and constant time distance decoder and routing decision.

(p,q)-planar graphs:

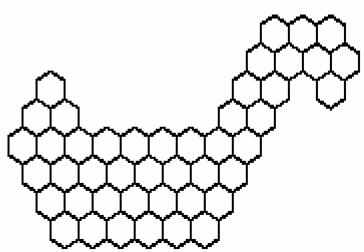
- inner faces of length at least p
- inner vertices of degree at least q



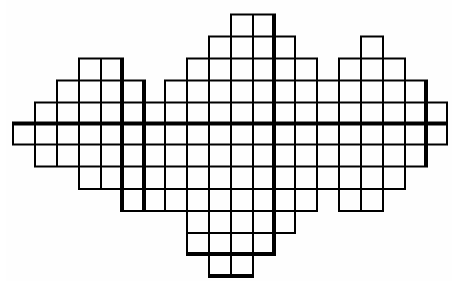
trigraph



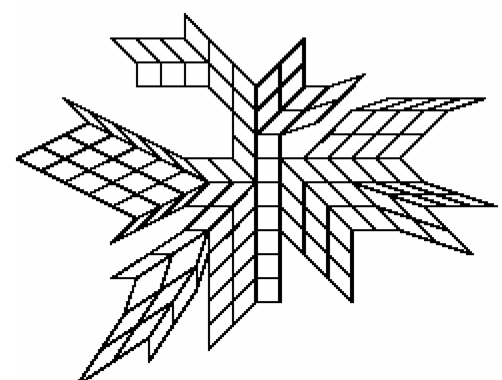
triangular system



hexagonal system



square system



squaregraph

Thank you