Connected Surveillance Game

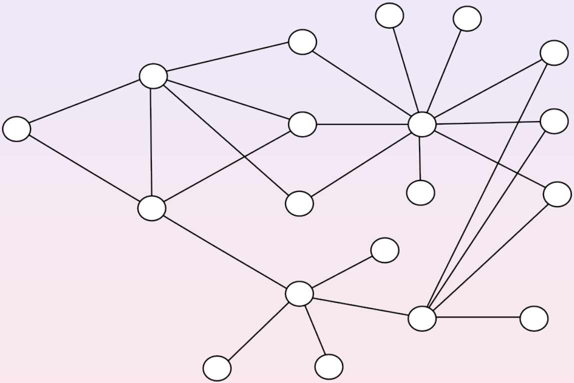
F. Giroire[1] D. Mazauric[2] N. Nisse[1] S. Pérennes[1]
R. P. Soares[1,3]

[1] COATI, Inria, I3S, CNRS, UNS, Sophia Antipolis, France
[2] ACRO, Laboratoire d'Informatique Fondamentale de Marseille, France
[3] ParGO Research Group, UFC, Fortaleza, Brazil

SIROCCO 2013

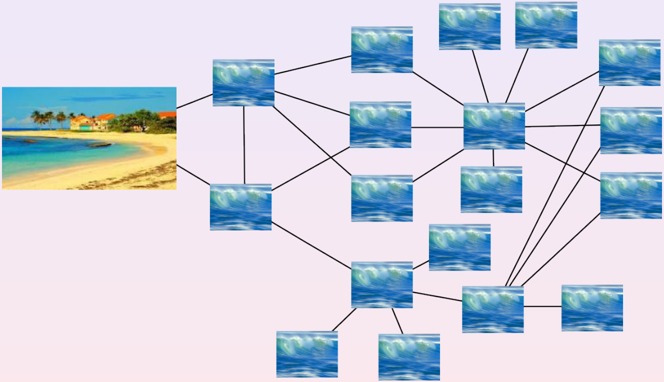
Ischia, July 1st, 2013

Surveillance game [Fomin,Giroire,Mazauric,Jean-Marie,Nisse 12]

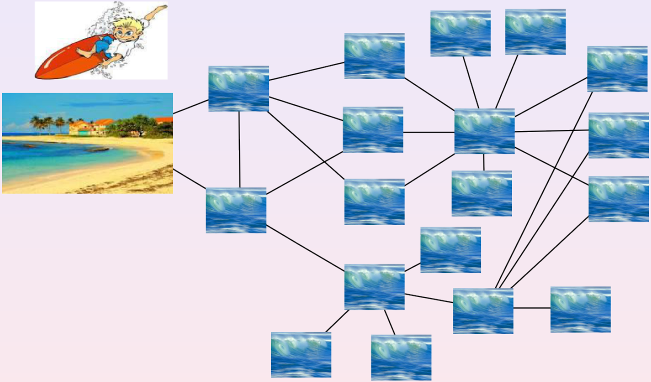


Two-Player game one a connected graph

Surveillance game [Fomin,Giroire,Mazauric,Jean-Marie,Nisse 12]

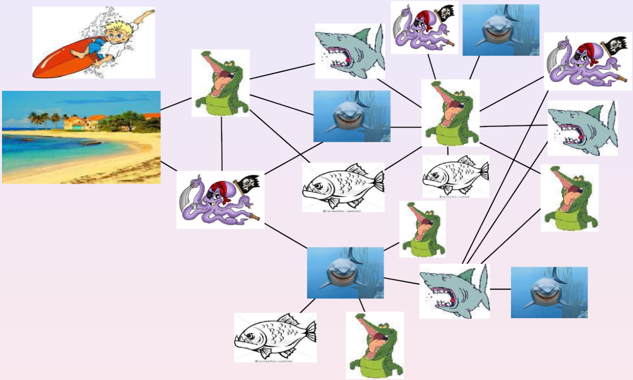


Two-Player game one a connected graph with given homebase



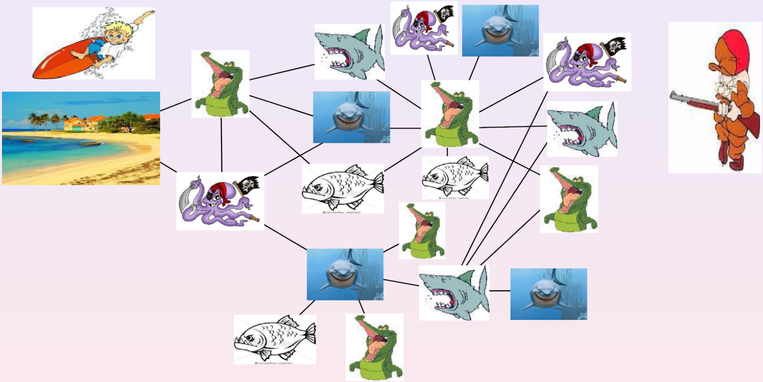
First Player: Surfer initially at the homebase

Surveillance game [Fomin,Giroire,Mazauric,Jean-Marie,Nisse 12]



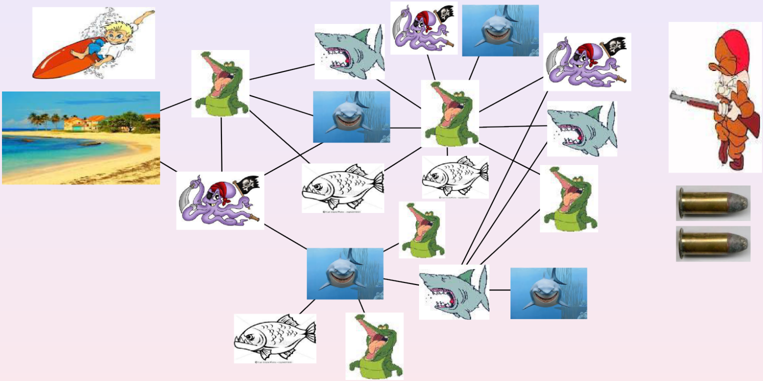
All nodes (but the homebase) are initially dangerous

Goal: avoid Surfer reaches one dangerous node



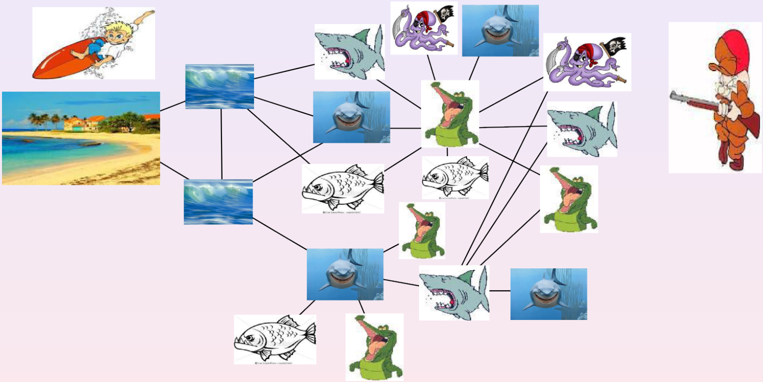
Second Player: Observer

Surveillance game [Fomin,Giroire,Mazauric,Jean-Marie,Nisse 12]



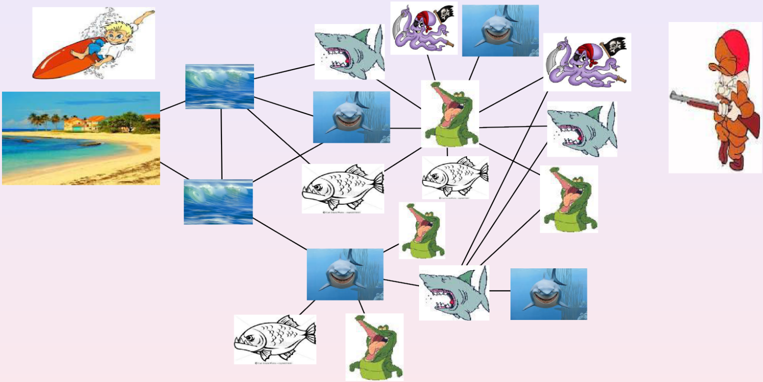
Observer: some amount of bullets (or marks) to secure nodes

Surveillance game [Fomin,Giroire,Mazauric,Jean-Marie,Nisse 12]

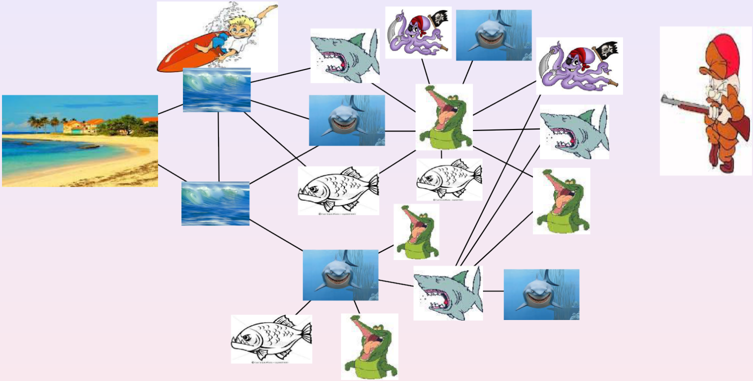


Observer turn: uses his bullets one bullet per node

Surveillance game [Fomin,Giroire,Mazauric,Jean-Marie,Nisse 12]



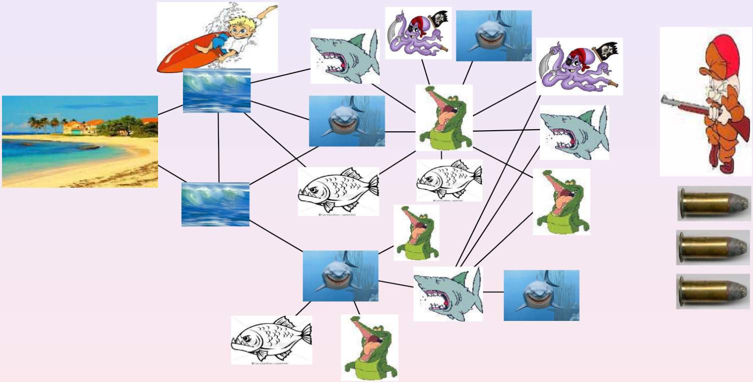
Observer turn: uses his bullets one bullet per node



Surfer turn: may move on adjacent node

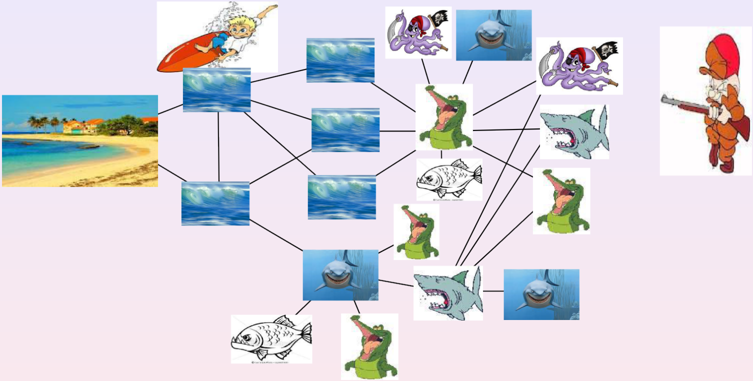
deg(*homebase*) $\leq$ # of bullets (per turn) required to save Surfer

Surveillance game [Fomin,Giroire,Mazauric,Jean-Marie,Nisse 12]



Naive strategy: mark all unmarked neighbors of current position

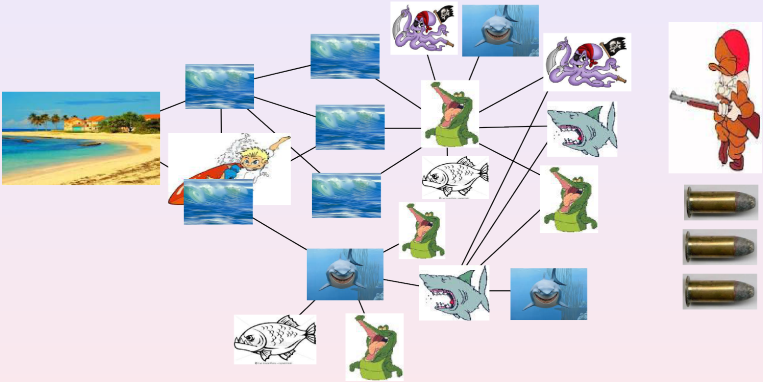
Surveillance game [Fomin,Giroire,Mazauric,Jean-Marie,Nisse 12]



Naive strategy: mark all unmarked neighbors of current position

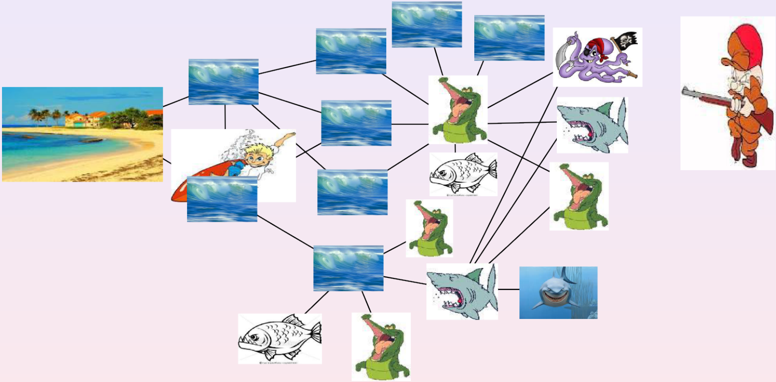
$$degree(homebase) \leq \text{amount of bullets} \leq max\ degree$$

Surveillance game [Fomin,Giroire,Mazauric,Jean-Marie,Nisse 12]



Surfer may move anywhere in its neighborhood

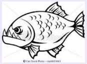
Surveillance game [Fomin,Giroire,Mazauric,Jean-Marie,Nisse 12]



bullets may be used to prevent future moves

Model: a Two players game

- a *Surfer*  starts from safe homebase v_0 

  in G, a dangerous graph     
- a *Guard*  with some amount k of bullets 

Turn by turn:

1. the guard secures $\leq k$ nodes;
2. then, the Surfer may move to an adjacent node.

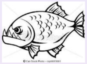
Defeat: Surfer in unsafe node **Victory:** G safe

Minimize amount of bullets to win for any Surfer's trajectory

Surveillance number of G (**connected**) from v_0: $sn(G, v_0)$

Model: a Two players game

- a *Surfer*  starts from safe homebase v_0 

  in G, a dangerous graph     
- a *Guard*  with some amount k of bullets 

Turn by turn:

1. the guard secures $\leq k$ nodes;
2. then, the Surfer may move to an adjacent node.

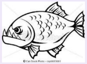
Defeat: Surfer in unsafe node  **Victory:** G safe 

Minimize amount of bullets to win for any Surfer's trajectory

Surveillance number of G (**connected**) from v_0: $sn(G, v_0)$

Model: a Two players game

- a *Surfer*  starts from safe homebase v_0 

  in G, a dangerous graph     

- a *Guard*  with some amount k of bullets 

Turn by turn:

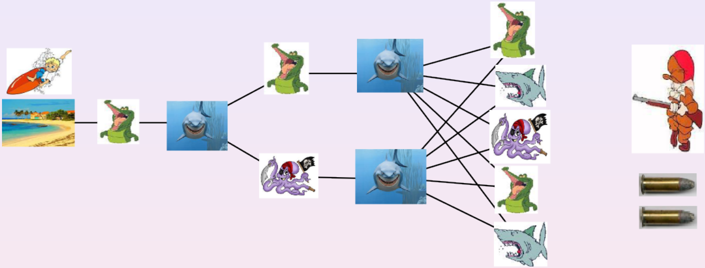
1. the guard secures $\leq k$ nodes;
2. then, the Surfer may move to an adjacent node.

Defeat: Surfer in unsafe node  **Victory:** G safe 

Minimize amount of bullets to win for any Surfer's trajectory

Surveillance number of G (**connected**) from v_0: $sn(G, v_0)$

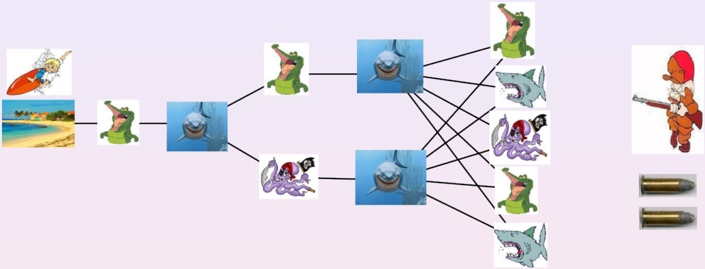
Simple Example



with 1 bullet: after 2 steps, Surfer faces 2 dangerous nodes!!

$$sn(G, v_0) > 1$$

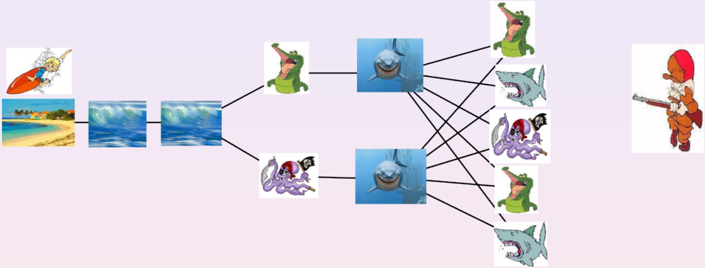
Simple Example



with 1 bullet: after 2 steps, Surfer faces 2 dangerous nodes!!

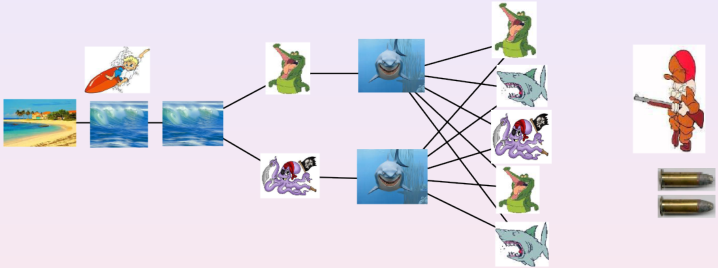
$$sn(G, v_0) > 1$$

Simple Example



Guard uses (all) his bullets

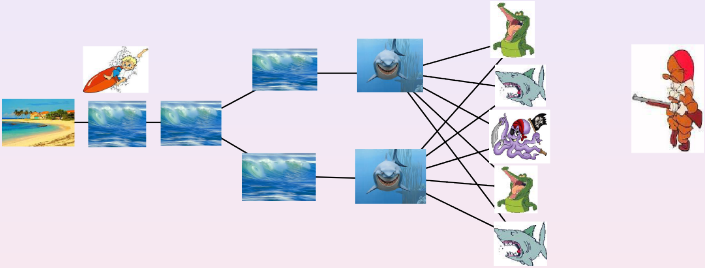
Simple Example



Guard uses (all) his bullets, **then** Surfer may move

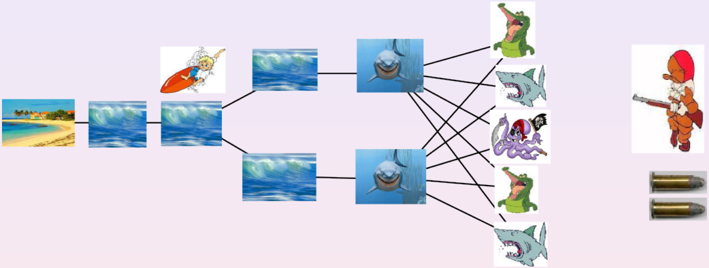
Clearly: worst case if Surfer always move

Simple Example



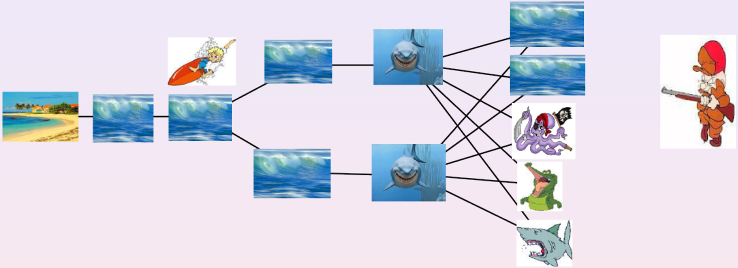
Guard uses (all) his bullets, **then** Surfer may move

Simple Example



Guard uses (all) his bullets, **then** Surfer moves

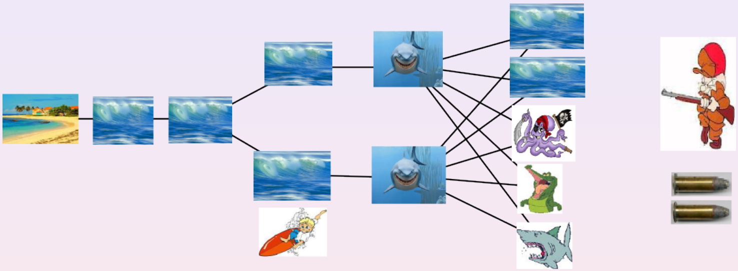
Simple Example



Guard uses (all) his bullets, **then** Surfer moves

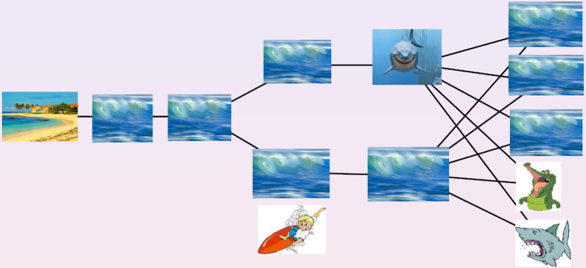
Guard may secure any node in the graph

Simple Example



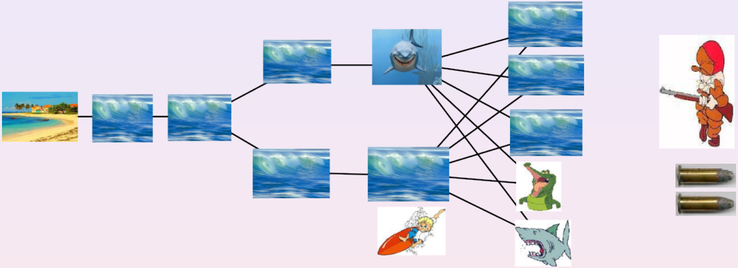
Guard uses (all) his bullets, **then** Surfer moves

Simple Example



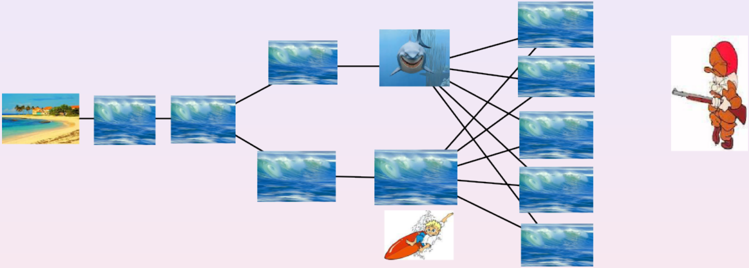
Guard uses (all) his bullets, **then** Surfer moves

Simple Example



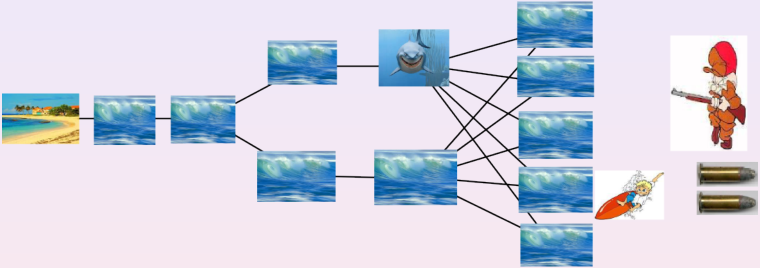
Guard uses (all) his bullets, **then** Surfer moves

Simple Example



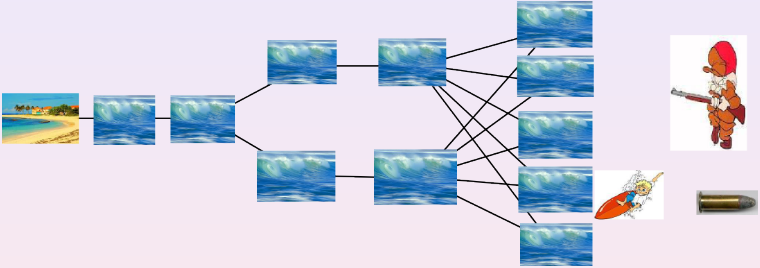
Guard uses (all) his bullets, **then** Surfer moves

Simple Example



Guard uses (all) his bullets, **then** Surfer moves

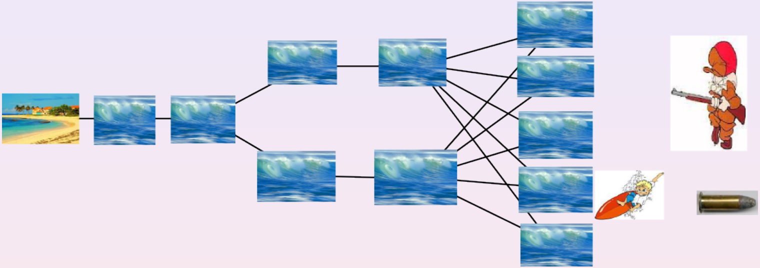
Simple Example



Guard uses (all) his bullets, **then** Surfer moves

All nodes safe: Victory against this trajectory of the Surfer

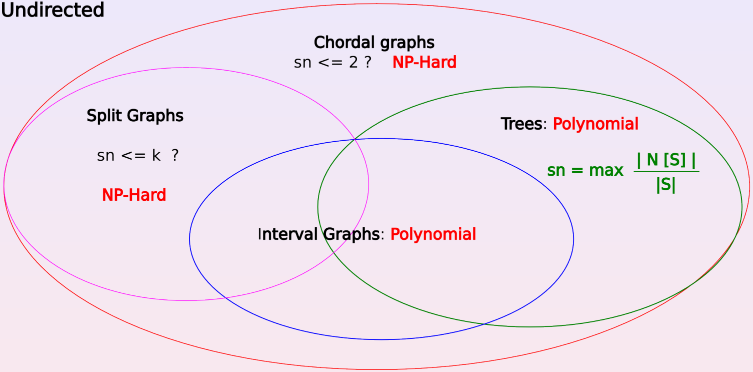
Simple Example



In this example, all Surfer's trajectory similar (by symmetry)

Victory whatever Surfer's trajectory $\Rightarrow sn(G, v_0) = 2$

Undirected



Directed

DAGs: sn <=4 ? **P-SPACE-Complete**

DAGs: sn <=2 ? **NP-Hard**

General

(out)-**degree (v0)** $\leq$ **sn** $\leq$ max (degree (v0) / **max degree** -1)

max out-degree

$O(2^n)$ exact algorithm

Motivations: model for prefetching problems

Web-page prefetching

Surfer $\Leftrightarrow$ Web-surfer following hyperlinks in the Web
Observer $\Leftrightarrow$ Web-browser downloading web-pages
Amount of bullets $\Leftrightarrow$ download speed
must be minimized (affect bandwidth)

Unrealistic assumptions

1. downloading Web-pages "far" from the current position of the Surfer
2. full knowledge of the Web-graph

To address 1^{st} point:

Connected Surveillance game [Fomin *et al.* 2012]

Motivations: model for prefetching problems

Web-page prefetching

Surfer ⇔ Web-surfer following hyperlinks in the Web
Observer ⇔ Web-browser downloading web-pages
Amount of bullets ⇔ download speed
must be minimized (affect bandwidth)

Unrealistic assumptions

1. downloading Web-pages "far" from the current position of the Surfer
2. full knowledge of the Web-graph

To address 1^{st} point:
Connected Surveillance game [Fomin *et al.* 2012]

Motivations: model for prefetching problems

Web-page prefetching

Surfer ⇔ Web-surfer following hyperlinks in the Web
Observer ⇔ Web-browser downloading web-pages
Amount of bullets ⇔ download speed
must be minimized (affect bandwidth)

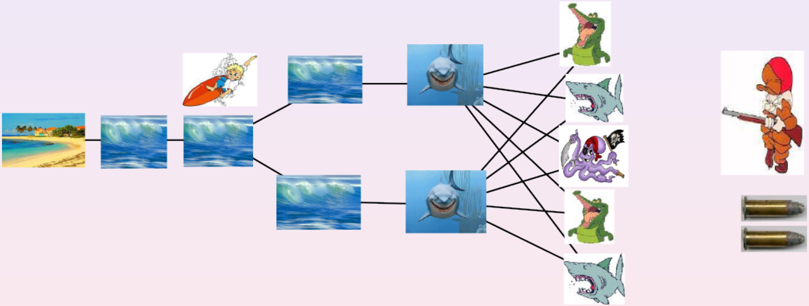
Unrealistic assumptions

1. downloading Web-pages "far" from the current position of the Surfer
2. full knowledge of the Web-graph

To address 1^{st} point:
Connected Surveillance game [Fomin *et al.* 2012]

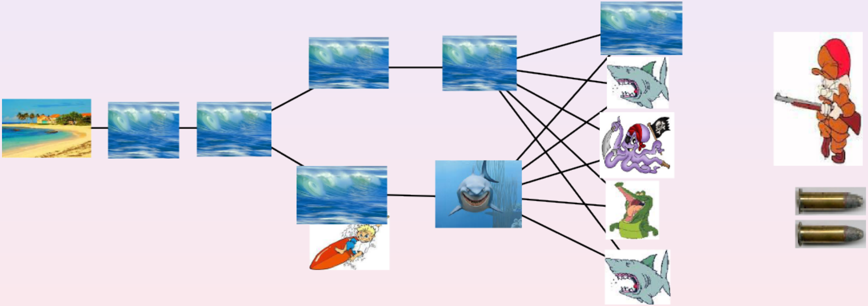
Connected Surveillance game

Constraint: safe vertices must induce a connected subgraph



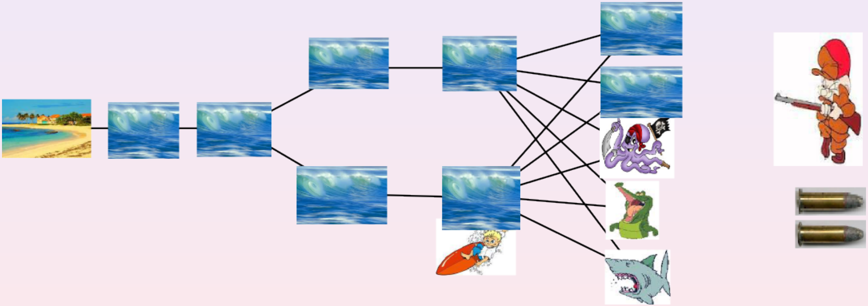
Connected surveillance number $csn(G)$

Constraint: safe vertices must induce a connected subgraph



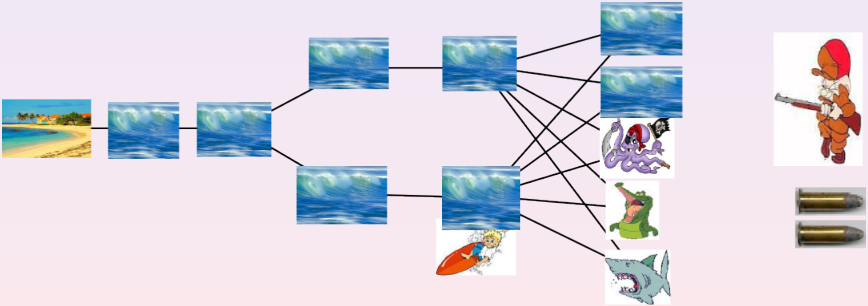
Connected surveillance number $csn(G)$

Constraint: safe vertices must induce a connected subgraph



Connected surveillance number $csn(G)$

Constraint: safe vertices must induce a connected subgraph



Connectivity costs: $csn(G, v_0) = 3 = sn(G, v_0) + 1$

Cost of connectivity and cost of "blindness"

Main question: $\exists$? $c > 1$ such that $\frac{csn(G)}{sn(G)} \leq c$ for any graph G?

Clearly $csn(G) \leq \Delta \cdot sn(G)$ for any G with max. degree Δ.

Connected Variant: the gap is still huge :(

Theorem 1 For any n-node graph $csn(G) \leq \sqrt{n \cdot sn(G)}$

Theorem 2 There are graphs G with $csn(G) = sn(G) + 2$

online variant: nodes are discovered when neighbor marked

restriction of the connected variant

Online variant: best online strategy is the naive one :(

Theorem 3 The best competitive ratio of online Protocol is $\Theta(\Delta)$.

Cost of connectivity and cost of "blindness"

Main question: $\exists ?\ c > 1$ such that $\frac{csn(G)}{sn(G)} \leq c$ for any graph G?

Clearly $csn(G) \leq \Delta \cdot sn(G)$ for any G with max. degree Δ.

Connected Variant: the gap is still huge :(

Theorem 1 For any n-node graph $csn(G) \leq \sqrt{n \cdot sn(G)}$

Theorem 2 There are graphs G with $csn(G) = sn(G) + 2$

online variant: nodes are discovered when neighbor marked

restriction of the connected variant

Online variant: best online strategy is the naive one :(

Theorem 3 The best competitive ratio of online Protocol is $\Theta(\Delta)$.

Cost of connectivity and cost of "blindness"

Main question: $\exists ?\ c > 1$ such that $\frac{csn(G)}{sn(G)} \leq c$ for any graph G?

Clearly $csn(G) \leq \Delta \cdot sn(G)$ for any G with max. degree Δ.

Connected Variant: the gap is still huge :(

Theorem 1 For any n-node graph $csn(G) \leq \sqrt{n \cdot sn(G)}$

Theorem 2 There are graphs G with $csn(G) = sn(G) + 2$

online variant: nodes are discovered when neighbor marked

restriction of the connected variant

Online variant: best online strategy is the naive one :(

Theorem 3 The best competitive ratio of online Protocol is $\Theta(\Delta)$.

Proof of Theorem 1

For any n-node graph $csn(G) \leq \sqrt{n \cdot sn(G)}$

Counting argument

1. Assume $sn(G) \leq k$
2. Connected strategy using $\sqrt{n \cdot k}$ marks per turn
   at each turn, mark $\sqrt{n \cdot k}$ unmarked neighbors of the position of the surfer (≈ naive strategy)
3. if Surfer wins after t turns at node v $\Rightarrow |N(v)| > \sqrt{nk}$
   - $t < \sqrt{n/k}$ since otherwise all nodes are marked
   - but $|N(v)| \leq k \cdot t$
     otherwise Surfer would have won in non-connected game
4. A contradiction

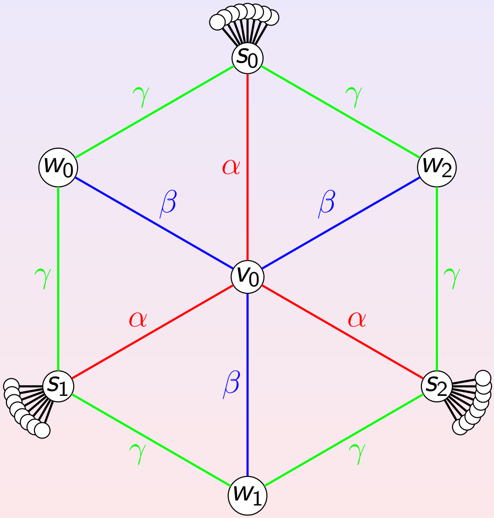
Sketch of Proof of Theorem 2

There are graphs G with $csn(G) = sn(G)+2$

What you don't see here

- paths of length α, β, γ (to be well whosen)
- all paths are doubled
- all nodes have "many" 1-degree neighbors
- in particular s_1, s_2, s_3

1.600.000 nodes



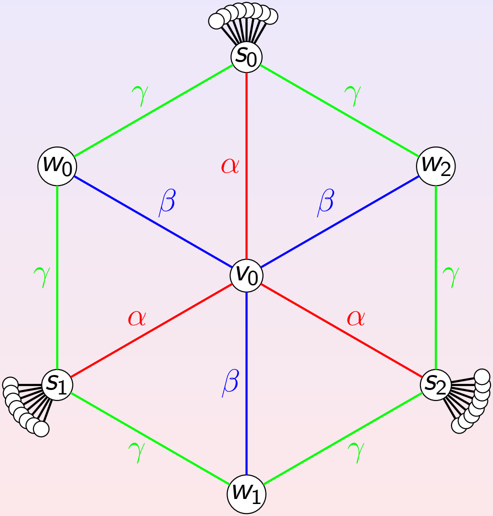
Sketch of Proof of Theorem 2

There are graphs G with $csn(G) = sn(G) + 2$

- Optimal non-connected strategy

  1^{st} step: mark "lot of"

  neighbors of s_1, s_2, s_3



Sketch of Proof of Theorem 2

There are graphs G with $csn(G) = sn(G)+2$

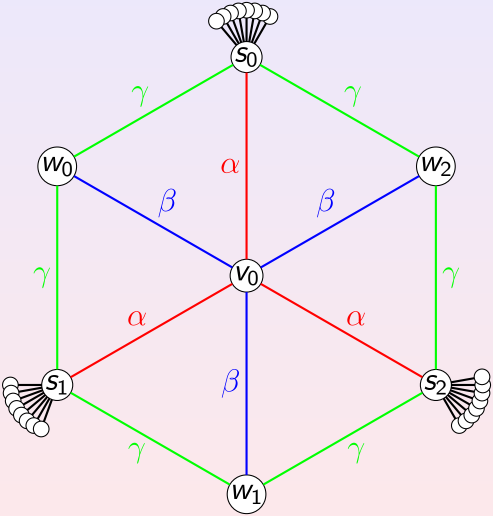
- Optimal non-connected strategy

  1^{st} step: mark "lot of"

  neighbors of s_1, s_2, s_3

- Connected strategies using $sn(G)+1$ bullets
  1^{st} step: mark many nodes "to reach" s_1, s_2, s_3 in

  $P_{v_0 s_1} \cup P_{v_0 w_2} \cup P_{w_2 s_0} \cup P_{w_2 s_2}$



Sketch of Proof of Theorem 2

There are graphs G with $csn(G) = sn(G)+2$

- Optimal non-connected strategy

  1^{st} step: mark "lot of"

  neighbors of s_1, s_2, s_3

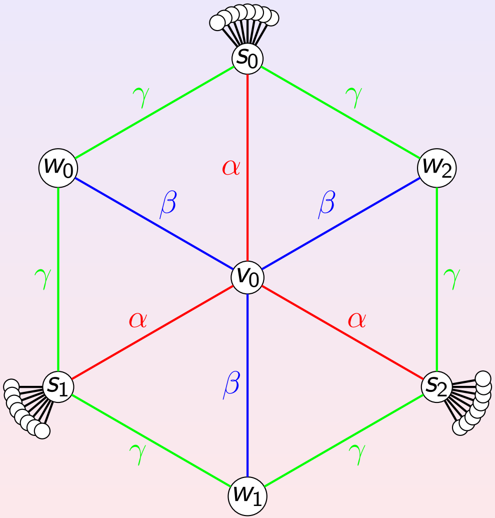
- Connected strategies using $sn(G)+1$ bullets
  1^{st} step: mark many nodes "to reach" s_1, s_2, s_3 in

  $P_{v_0 s_1} \cup P_{v_0 w_2} \cup P_{w_2 s_0} \cup P_{w_2 s_2}$

- Surfer goes along $P'_{v_0 w_2}$

  one more bullet not enough

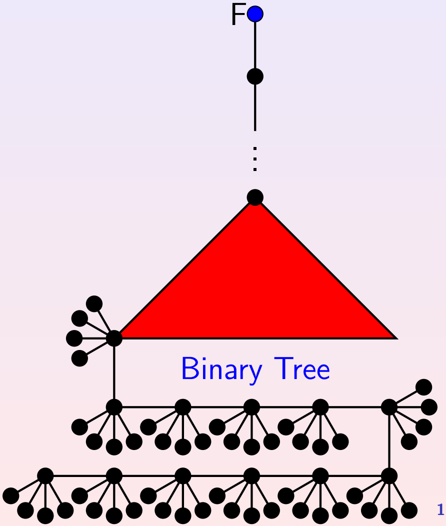
  to protect s_0 and s_2



Sketch of Proof of Theorem 3

Online competitive ratio: $\Theta(\Delta)$

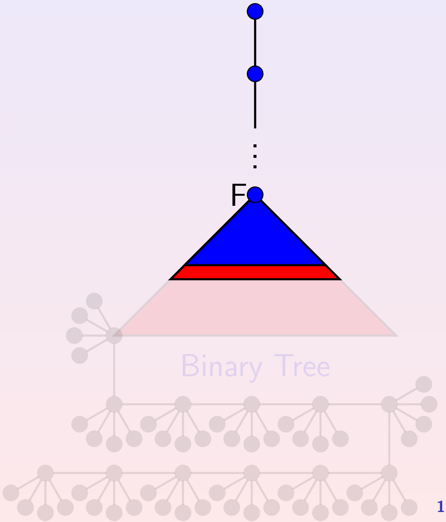
- A tree T with $sn(T) = 2$
  full knowledge: Observer can anticipate the heavy branch



Sketch of Proof of Theorem 3

Online competitive ratio: $\Theta(\Delta)$

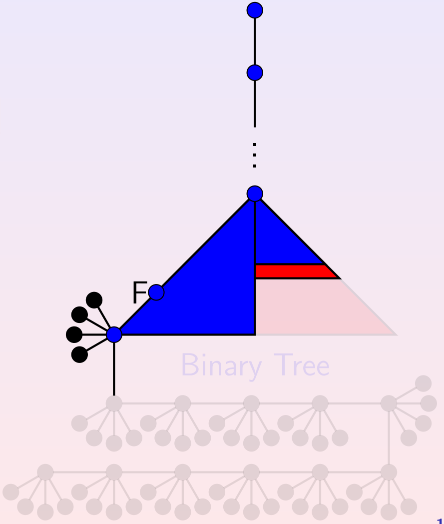
- A tree T with $sn(T) = 2$
  full knowledge: Observer can anticipate the heavy branch
- online: symmetrical view
  cannot anticipate



Sketch of Proof of Theorem 3

Online competitive ratio: $\Theta(\Delta)$

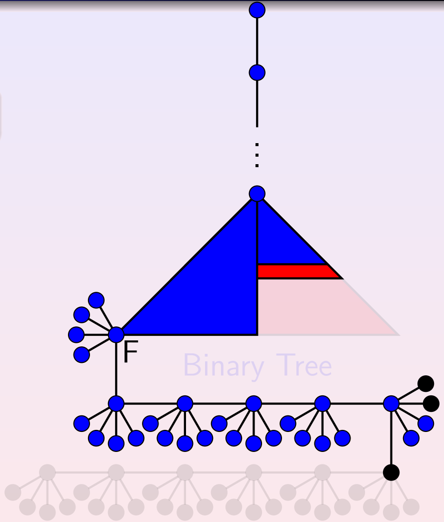
- A tree T with $sn(T) = 2$
  full knowledge: Observer can anticipate the heavy branch
- online: symmetrical view
  cannot anticipate
- when the heavy branch appears



Sketch of Proof of Theorem 3

Online competitive ratio: $\Theta(\Delta)$

- A tree T with $sn(T) = 2$
  full knowledge: Observer can anticipate the heavy branch
- online: symmetrical view
  cannot anticipate
- when the heavy branch appears
- ...it is too late if Observer uses only $o(\Delta)$ bullets



Open Questions

- complexity in bounded degree graphs?
  (polynomial if $\Delta \leq 3$)
- complexity in bounded treewidth graphs?
- $\exists ? c < 2$ and $O(c^n)$ algorithm in n-node graphs?
- cost of connectivity? $\frac{csn}{sn} \leq cte$?
- More realistic model: finite memory
  what if nodes may be "recontaminated"?

Thank you