Network Anti-Inference: A Fundamental Perspective on Proactive Strategies to Counter Flow Inference

Zhuo Lu
University of Memphis

Cliff Wang
Army Research Office

Outline

- Network inference
- Network anti-inference
  - Deception traffic
  - Routing changing
- Analysis and examples
- Simulation results
- Conclusions

Network Inference

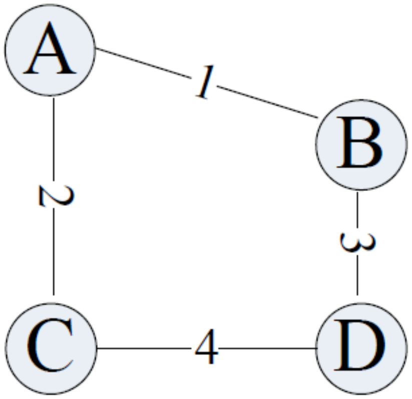
- Also called network tomography
  - Building a relationship between link and flow information. Then, Inferring one from the other.
    - Given link rate info, get the flow rate info;
    - Given flow rate info, get the link rate info;
- Applications: fault diagnose, network monitoring, flow detection, …
- We focus on flow inference in wireless networks.
  - Goal: make flow inference inaccurate, which is called anti-inference!

Inference: Problem Formulation

- Flow inference formulation: $y = Ax$
  - y – link rate vector: observed by attackers
  - x – flow rate vector: to be estimated
  - A – routing matrix: known network info
- Given A and y, estimate x
  - Usually an under-determined system
  - So no least squares solution!

How to Get Routing Matrix A

- Example:



link indexing: 1-4

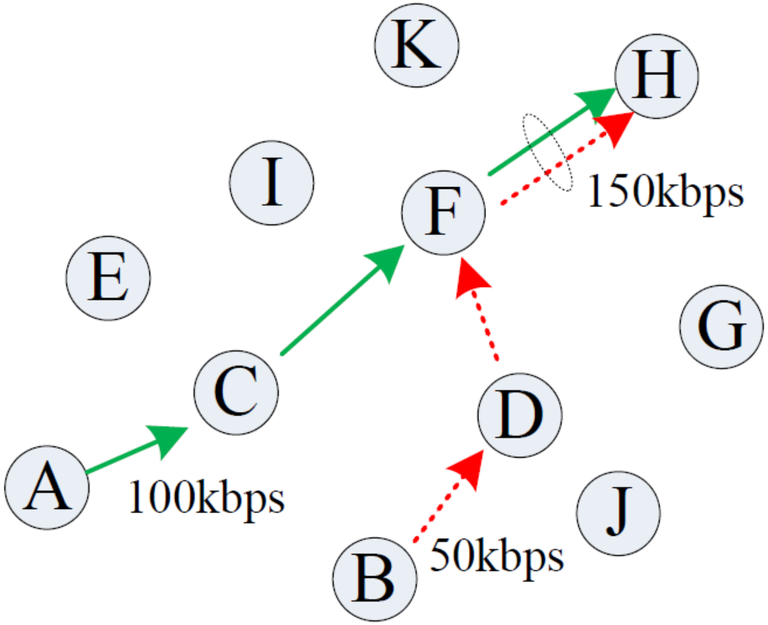
flow indexing: 1-6

$$\begin{array}{cc|cccccc}
 & & \text{(A-B)} & \text{(A-C)} & \text{(A-D)} & \text{(B-C)} & \text{(B-D)} & \text{(C-D)} \\
 & & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
\text{(A-B)} & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
\text{(A-C)} & 2 & 0 & 1 & \underline{a_{2,3}=1} & 0 & 0 & 0 \\
\text{(B-D)} & 3 & 0 & 0 & 0 & 1 & 1 & 0 \\
\text{(C-D)} & 4 & 0 & 0 & \underline{a_{4,3}=1} & 1 & 0 & 1
\end{array}$$

routing matrix **A**

Example

- Observing link transmissions (knowing y)
  - 11 nodes, 2 flows, y=Ax ➔ get x from y.
  - Inference Result: A→H: 100kbps, B→H: 50kbps

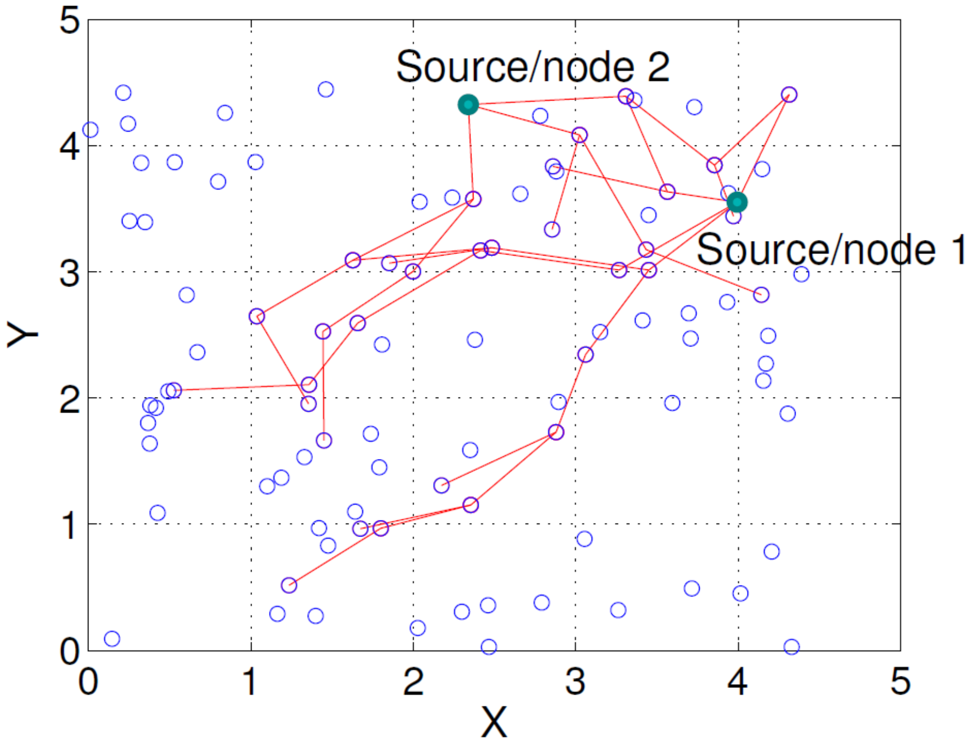


Network Inference: Negative Side

- Network inference:
  - Get some information by observing.

Example:

- Two critical nodes are multicasting info in the network,
- By using network inference, an adversary can infer all network flows by observing link transmission.
  - Know who are critical nodes.

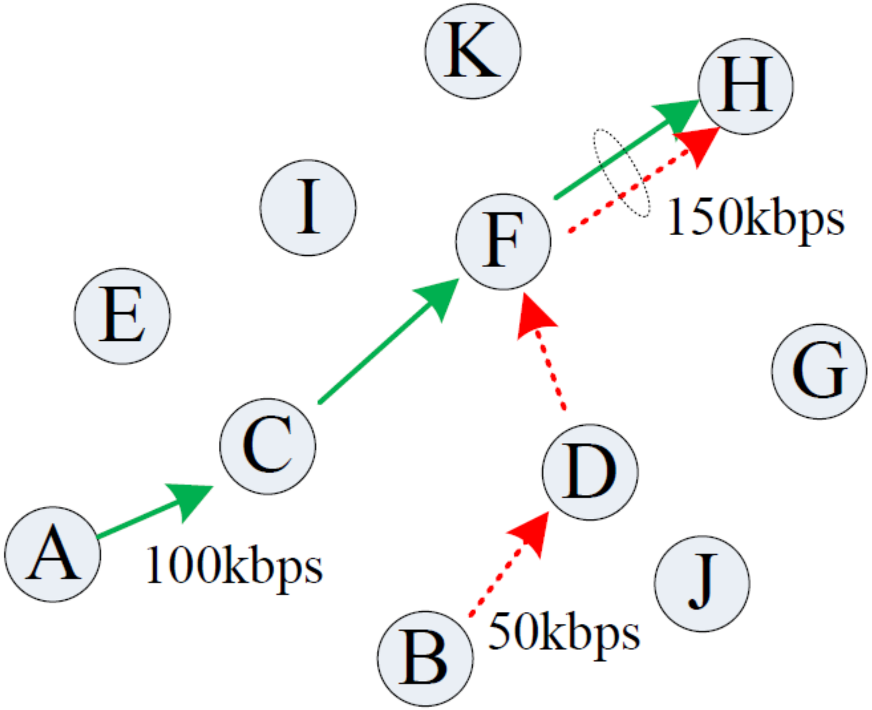


Network Anti-Inference

- Definition:
  - Methods that make network inference inaccurate!
- Attacker:
  - Try to infer the rate of all network flows by observing link transmissions.
- Our objective is to answer:
  - What are the possible methods?
  - What is the benefit?
  - What is the cost?

How to break inference?

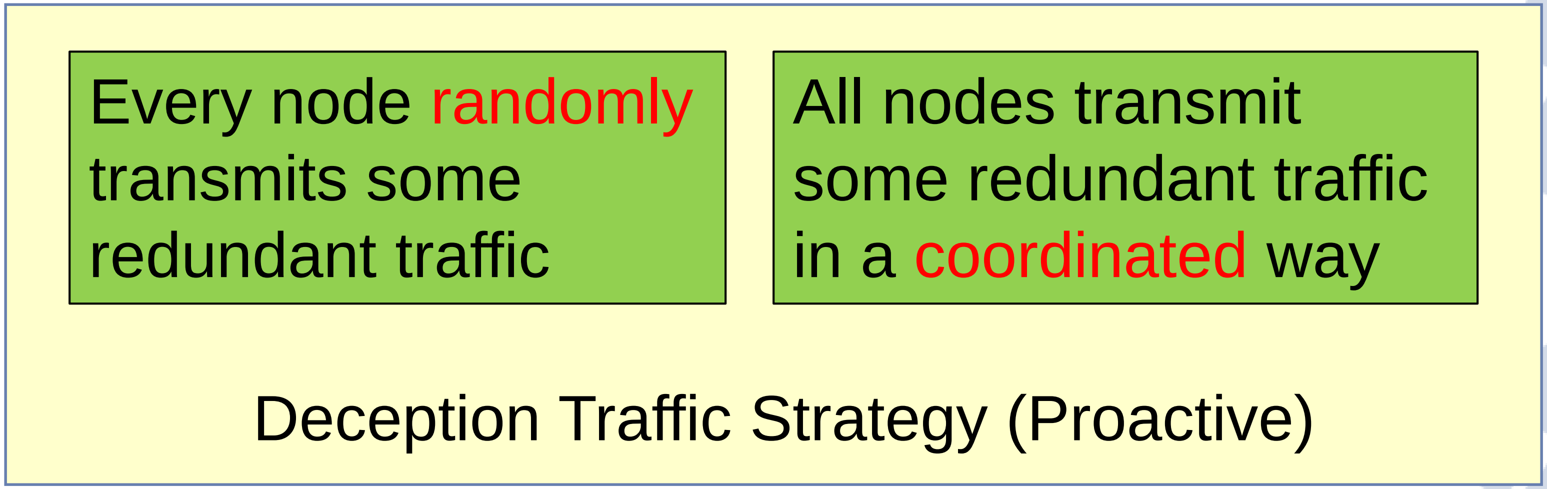
- Two underlying assumptions for inference
  - Link traffic is only induced by network flows
    - No flow → no link traffic
  - Routing is usually predictable
    - E.g., shortest path routing.



Anti-inference: break at least one of these assumptions!
We have to be proactive!

Deception Traffic

- Link traffic is only induced by network flows
  - No flow → no link traffic



Every node randomly transmits some redundant traffic

All nodes transmit some redundant traffic in a coordinated way

Deception Traffic Strategy (Proactive)

Routing Changing

- Routing is usually predictable
  - E.g., shortest path routing.

Dynamically change routing paths to make sure the attacker has some information mismatch

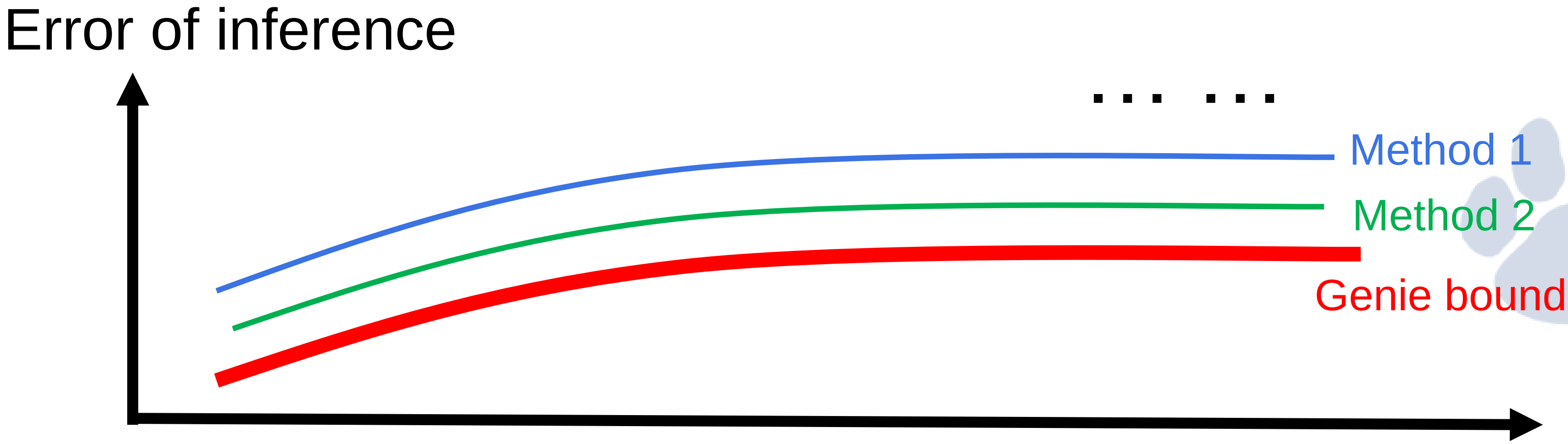
Routing Changing Strategy (Proactive)

Formulation for Anti-Inference

- Original formulation:
  - $y = Ax$
- Deception Traffic:
  - Add noise: $y = Ax + J$ (← deception traffic vector)
- Routing Changing:
  - Information mismatch: changing routing means routing matrix A → B (← new routing matrix)

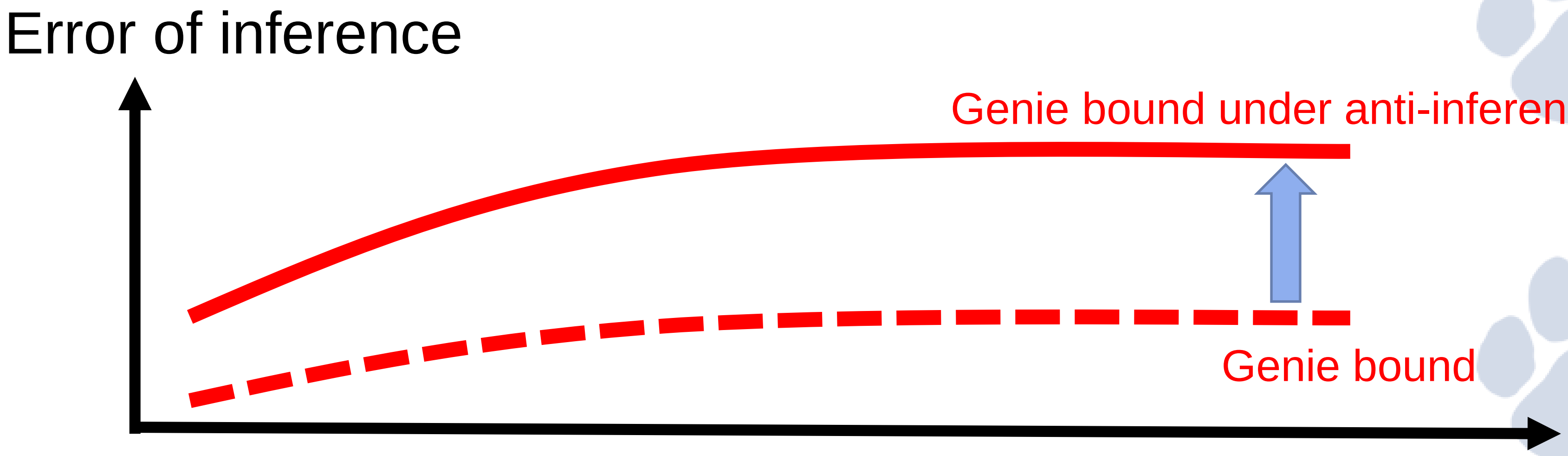
Metric to Measure the Benefit

- Metrics to measure the accuracy of network inference? Genie bound: lower bound of error in all possible methods.
    - Assuming the attacker knows who is transmitting,
    - Then using minimum mean squared error estimation to estimate all the flow rates.



Genie Bound

- We want to see how much the genie bound can be increased due to deception traffic and routing changing with bounded costs.



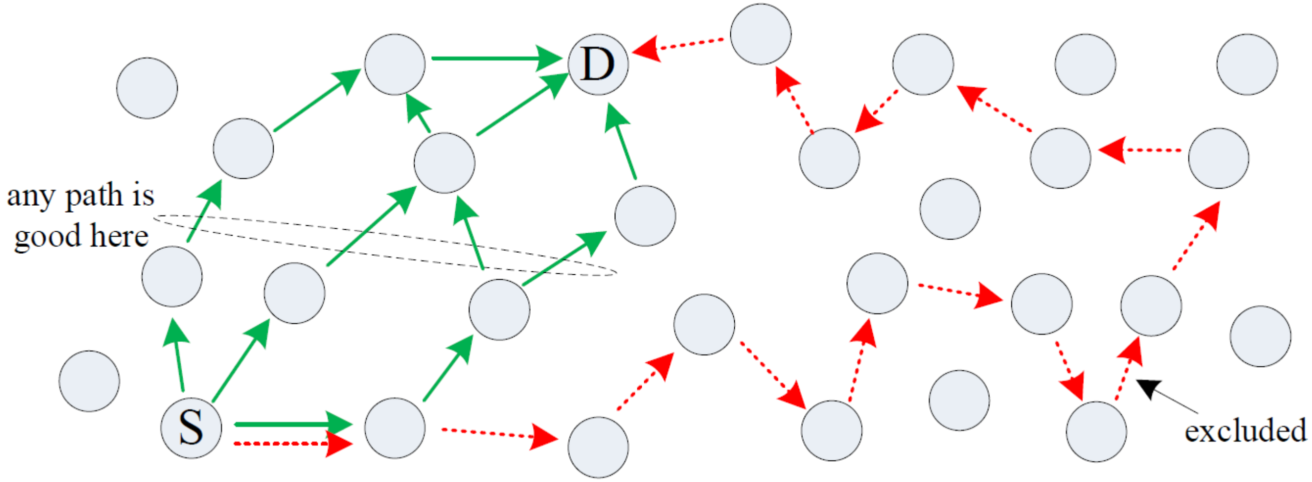
Bound the Costs

- Deception Traffic: $y = Ax + J$
  - $|J|/n$, or $E|J|/n$ (average deception traffic per node) is smaller than a constant, where *n* is the number of nodes in the network.

- Routing Changing: A → B
  - We have a random geometric graph model, all nodes are randomly distributed.
  - A and B are random matrices.
  - How to model the routing changing ??

Routing Modeling

- Model: Under any routing strategy, the average number of hops between any source-destination pair is denoted by a function $g(n)$ satisfying $g(n) = O(n)$, where n is the number of nodes in the network



  - Existing K-shortest path routing satisfies this model.

Routing Modeling II

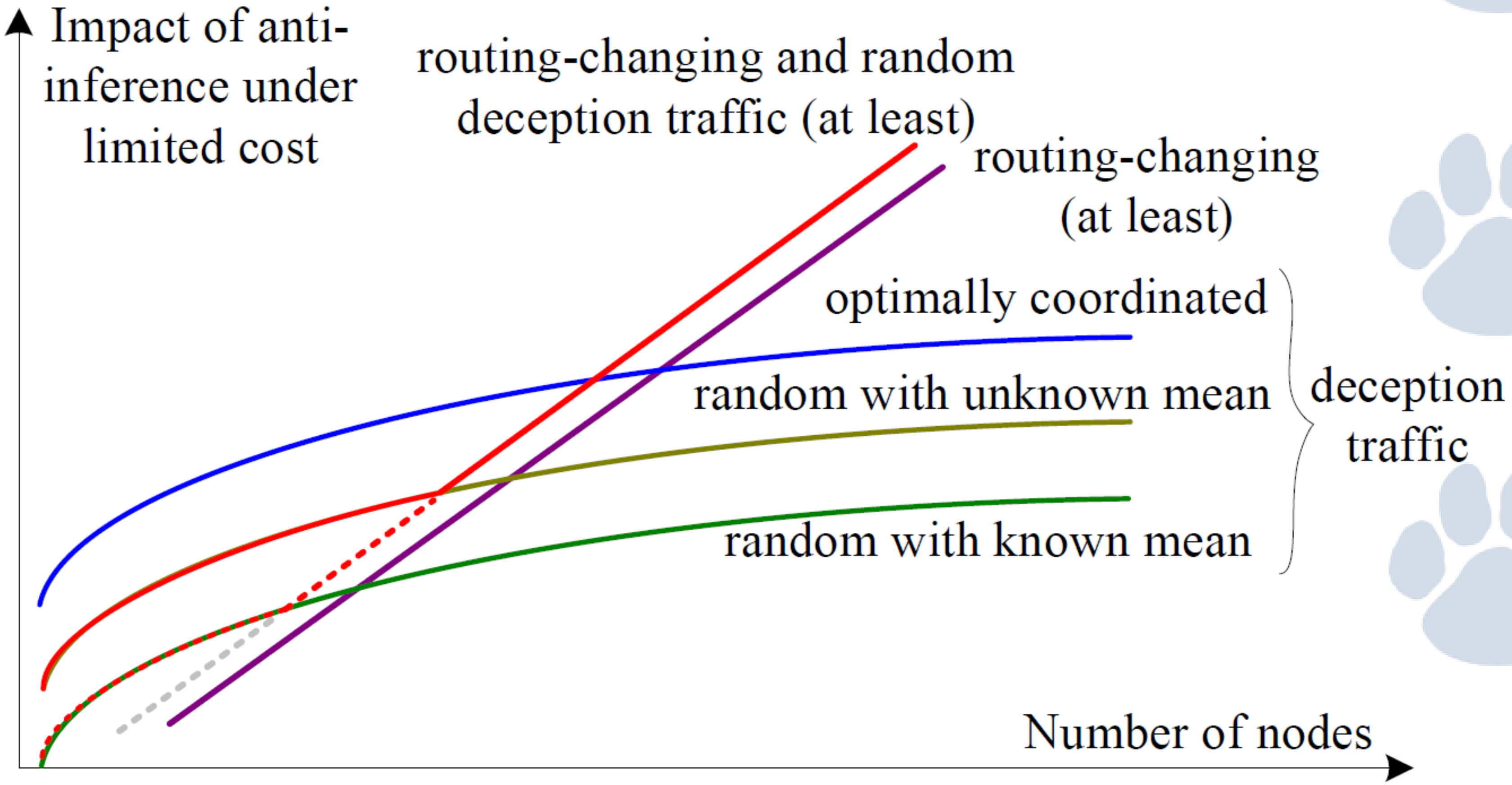
- Quantifying the cost of routing changing:
  - The original routing changing: $g(n)$
  - The new routing changing: $h(n)$
  - The cost is $h(n)/g(n)$,

  where n is the number of nodes in the network.

  Limit the cost: $\Theta(h(n)/g(n)) = \Theta(1)$,

Theoretical Result: An Example

- In a network with *n* nodes, Θ(n) random network flows.



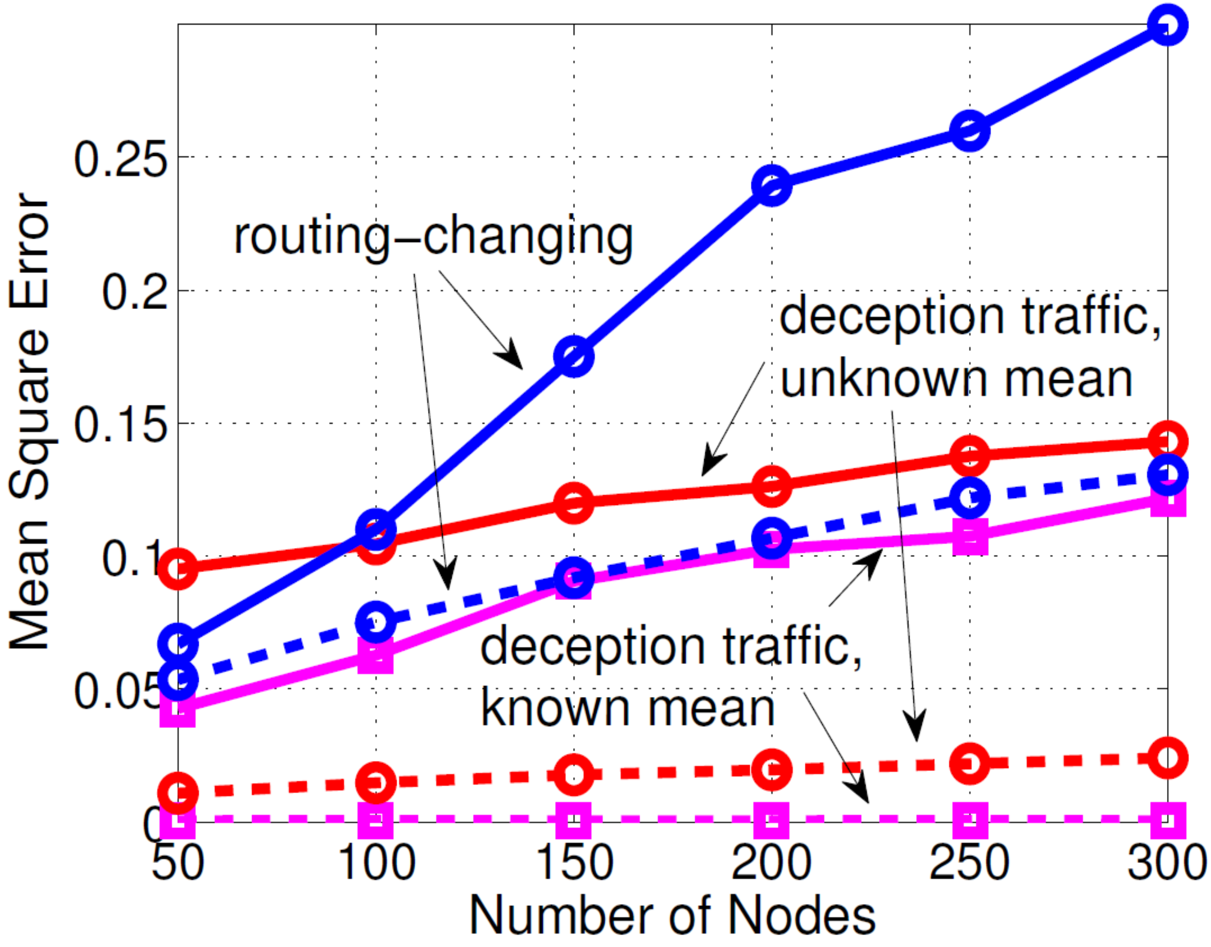
Simulation Results

Inference:

- in-crowd algorithm (Gill, *et al*, 2011) for inference

Anti-inference

- ~50% deception traffic in the network,
- ~30% hop increase in routing changing



Dashed lines – Genie bounds; Solid slides – MSEs of in-crowd

Conclusions

- Network anti-inference
- A fundamental view on proactive strategies:
  - Deception traffic
  - Routing changing

- Random traffic has the impact on the same order of the best coordinated traffic.
- Routing changing is generally better than the deception traffic.

Thank you!

Q/A?