Game theoretical approaches to secure and robust routing

João P. Hespanha

Center for Control Engineering and Computation

University of California
Santa Barbara

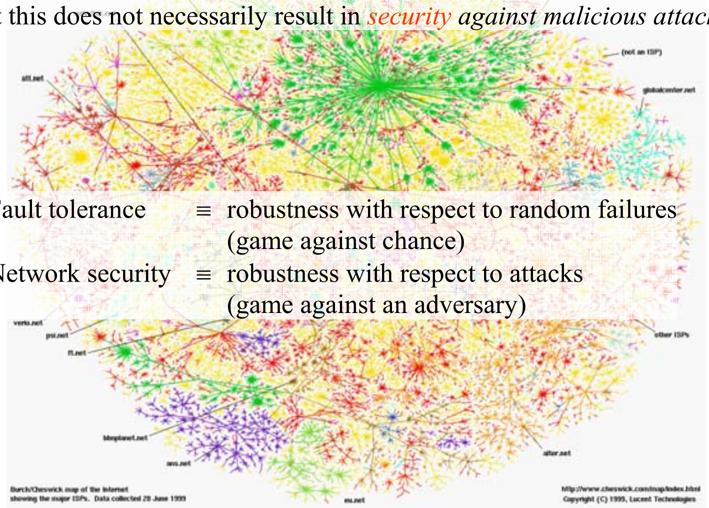
UCSB

In collaboration with: S. Bohacek (Univ. Delaware), K. Obraczka (UC Santa Cruz)
J. Lee (Postdoc, UC Santa Barbara), C. Lim (PhD candidate, USC)

Network Security vs. Fault-Tolerance

The basic principle behind the design of the Internet was to utilize massive redundancy to achieve fault-tolerance

but this does not necessarily result in *security against malicious attacks*



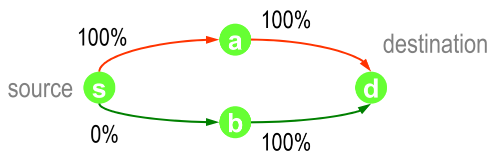
Fault tolerance ≡ robustness with respect to random failures (game against chance)

Network security ≡ robustness with respect to attacks (game against an adversary)

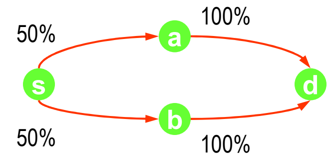
An adversary can explore weaknesses that chance will not easily find

Security vs. Fault-Tolerance in Routing

single-path routing



stochastic multi-path routing



Suppose all links are equally likely to fail, and one of them does fail…

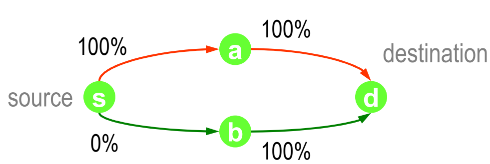
Which routing strategy results in higher probability that a packet will reach destination?

link labels refer to probability of forwarding a packet

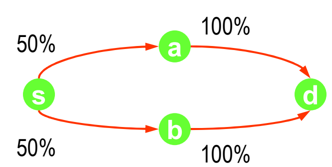
Both routing schemes result in exactly the same probability (50%)…

Security vs. Fault-Tolerance in Routing

single-path routing



stochastic multi-path routing



Suppose all links are equally likely to fail, and one of them does fail…

With a transport protocol that guarantees reliable delivery (e.g., TCP):

$$\mathrm{E}\Big[\#\ \text{transmissions}\Big] = 50\% \times 1 + 50\% \times \infty = \infty$$

$$\mathrm{E}\Big[\#\ \text{transmissions}\Big] = \sum_{i=1}^{\infty} (1+i)(.50)^i .50 = 2$$

prob. of failing i times and succeeding at next attempt

Security vs. Fault-Tolerance in Routing

Disclaimers:

1. Eventually routing would get fixed but 40% of the path outages take more than 30 minutes to repair [Chandra et al., 2003]
2. If failure is bidirectional, acknowledgement packets can also be dropped and E[# transmissions] raises to 4

With a transport protocol that guarantees reliable delivery (e.g., TCP):

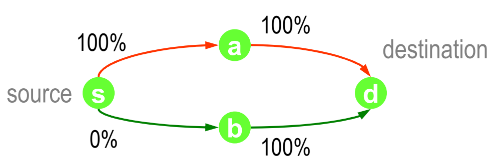
$$\mathrm{E}\left[\#\ \text{transmissions}\right] = 50\% \times 1 + 50\% \times \infty = \infty$$

$$\mathrm{E}\left[\#\ \text{transmissions}\right] = \sum_{i=1}^{\infty}(1+i)(.50)^i.50 = 2$$

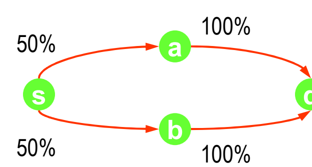
prob. of failing i times and succeeding at next attempt

Security vs. Fault-Tolerance in Routing

single-path routing



stochastic multi-path routing



~~Suppose all links are equally likely to fail~~, and one of them does fail…

Assume that fail was caused by an attacker that selects the link

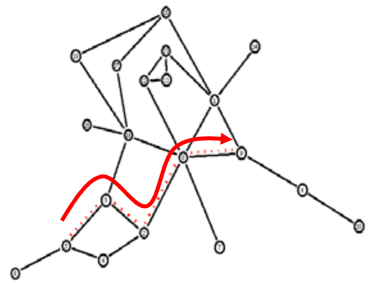
Which routing strategy results in higher probability that a packet will reach destination?

Attacker can learn routing policy and prevent all communication by compromising a single link

Compromising a single link, probability of intercepting packet is only 50% (assuming stochastic multi-path)

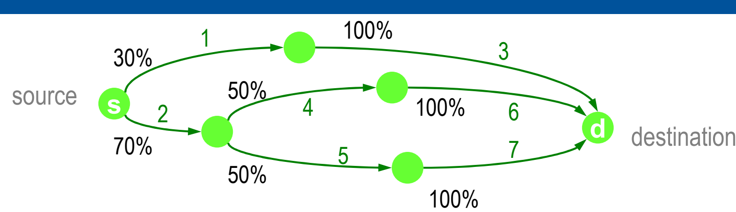
later we will find other reasons why multi-path may be advantageous…

Outline



1. How to compute stochastic multi-path routing tables?
   Noncooperative game—explore redundancy in an adversarial context
   - Different levels of security/reliability for distinct links/nodes
   - Reduce latency
2. Other issues:
   - Scalable computation for large networks
   - Multi-path routing within the protocol stack

Stochastic routing policies



probability that a packet arriving at the node where ℓ starts will be routed though link ℓ

set of (unidirectional) links

stochastic routing policy $\equiv R := \{\, r_\ell \geq 0 : \ell \in \mathcal{L} \,\}$

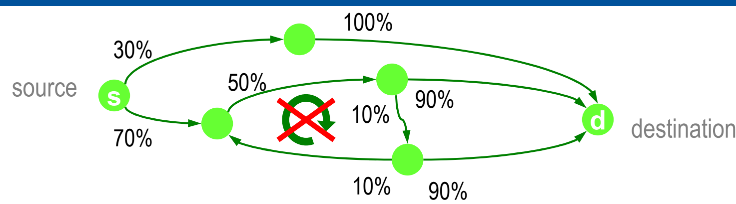
for every node n $\displaystyle\sum_{\ell \in \mathcal{L}[n]} r_\ell = 1$

summation over links that exit node n

e.g., $R := \{\, \underbrace{.3, .7}_{\text{add to 1}}, 1, \underbrace{.5, .5}_{\text{add to 1}}, 1, 1\}$

$\mathcal{R}_{\text{stoch}} \equiv$ set of all routing policies

Stochastic routing policies



probability that a packet arriving at the node where ℓ starts will be routed though link ℓ

set of (unidirectional) links

stochastic routing policy $\equiv R := \{\, r_\ell \geq 0 : \ell \in \mathcal{L} \,\}$

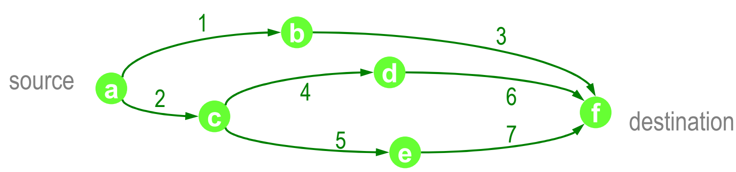
for every node n $\displaystyle\sum_{\ell \in \mathcal{L}[n]} r_\ell = 1$

summation over links that exit node n

$\mathcal{R}_{\text{stoch}}$ $\equiv$ set of all routing policies

$\mathcal{R}_{\text{no-cycle}}$ $\equiv$ set of all cycle-free policies, i.e., for which there is no closed sequence of links all with positive routing probability

Attack space



probability that packets in link ℓ are compromised

set of links

pure attack $\equiv$ $P := \{\, p_\ell : \ell \in \mathcal{L} \,\}$

attacker has available a pool of "pure attacks" and will select the one that is more likely to prevent communication

e.g., pure attack at link 3 with 10% probability of success:

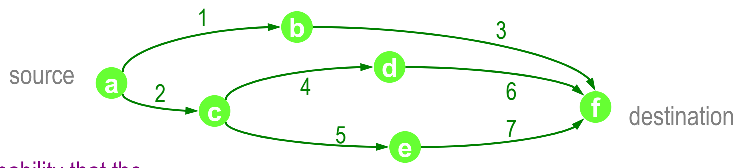
$P_3 := \{\, 0, 0, .1, 0, 0, 0 \,\}$

pure attack at node f with 20% probability of success:

$P_{\text{f}} := \{\, 0, 0, .2, 0, .2, .2 \,\}$

$\mathcal{P} \equiv$ set of all (pure) attacks available to attacker

Mixed attacks



probability that the attacker will intercept a packet traveling in link ℓ

set of links

(pure) attack $\equiv \quad P := \{\, p_\ell : \ell \in \mathcal{L} \,\}$

attacker is allowed to randomize between pure attacks with appropriate probabilities

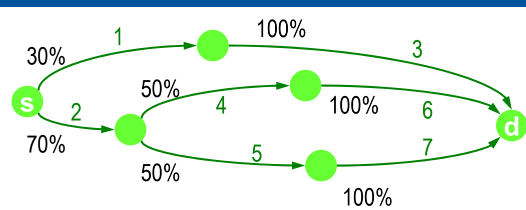
$\mathcal{P} \equiv$ set of all pure attacks available to attacker

mixed attack policy $\equiv M := \{\, m_P : P \in \mathcal{P} \,\} \in [0,1]^{\mathcal{P}}$

$$\sum_{P \in \mathcal{P}} m_P = 1$$

probability that the attacker select the pure attack P

Example



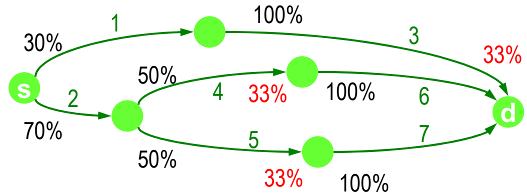
stochastic routing policy $\equiv R := \{ .3, .7, 1, .5, .5, 1, 1\}$

pure attacks available to attacker $\equiv \mathcal{P} := \{ \{.1,0,0,0,0,0,0\},\{0,.1,0,0,0,0,0\}, \ldots$
$\ldots,\{0,0,0,0,0,.1,0\}, \{0,0,0,0,0,0,.1\} \}$

10% effective link attacks (7 attacks)
(attacker can target any link, it will succeed in compromising packet delivery with 10% probability)

Example

UCSB



stochastic routing policy $\equiv R \coloneqq \{ .3, .7, 1, .5, .5, 1, 1\}$

pure attacks available to attacker $\equiv \mathcal{P} \coloneqq \{ \{.1,0,0,0,0,0,0\},\{0,.1,0,0,0,0,0\}, \dots$
$\dots,\{0,0,0,0,0,.1,0\}, \{0,0,0,0,0,0,.1\} \}$

mixed attack policy $\equiv M \coloneqq \{ 0, 0, .33, .33, .33, 0, 0 \}$

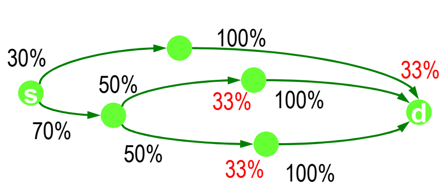
$$\mathrm{P}_{R,M}(\text{capture}) = \big(30\% \times 33\% + 2 \times 35\% \times 33\%\big) \times 10\% = 3.3\%$$

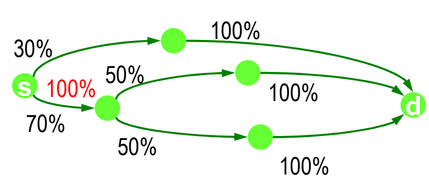
probability that packet is captured for routing policy R and mixed attack policy M

but not really rational…

Example

UCSB





mixed attack policy
$M \coloneqq \{ 0, 0, .33, .33, .33, 0, 0 \}$

stochastic routing policy
$R \coloneqq \{ .3, .7, 1, .5, .5, 1, 1\}$

$$\mathrm{P}_{R,M}(\text{capture}) = \big(30\% \times 33\% + 2 \times 35\% \times 33\%\big) \times 10\% = 3.3\%$$

for this attack policy M
router cannot do better

$$\mathrm{P}_{R,M}(\text{capture}) = 3.3\% \quad \forall R$$

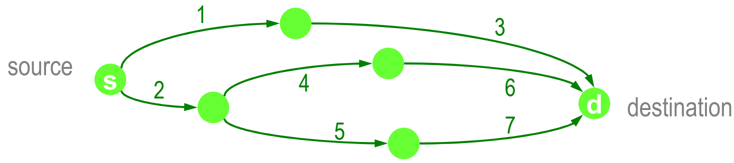
attacker could do better against R with
$M \coloneqq \{ 0, 1, 0, 0, 0, 0, 0 \}$

$$\mathrm{P}_{R,M}(\text{capture}) = \big(70\% \times 100\%\big) \times 10\% = 7\%$$

but then …

neither of the above policies is an "equilibrium" since at least one player can improve its outcome by changing its policy

Routing game



Compute saddle-point equilibrium policies:

$R^* \in \mathcal{R}_{\text{no-cycle}}$ (cycle-free stochastic routing policy)
$M^* \in [0,1]^{\mathcal{P}}$ (mixed attack policy)

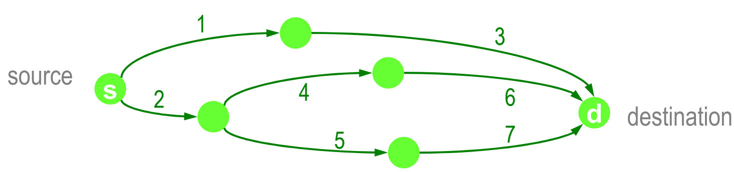
for which

$$\begin{aligned} \mathrm{P}_{R^*,M^*}(\text{capture}) &= \min_{R\in\mathcal{R}_{\text{no-cycle}}} \max_{M\in[0,1]^{\mathcal{P}}} \mathrm{P}_{R,M}(\text{capture}) \\ &= \max_{M\in[0,1]^{\mathcal{P}}} \min_{R\in\mathcal{R}_{\text{no-cycle}}} \mathrm{P}_{R,M}(\text{capture}) \end{aligned}$$

Existence?
Computation?

policies chosen by intelligent opponents to minimize their worst-case losses
(no player will improve its outcome by deviating from equilibrium)

Probability of capture



Given

$R \in \mathcal{R}_{\text{no-cycle}}$ (cycle-free stochastic routing policy)
$M := \{ m_P : P \in \mathcal{P} \} \in [0,1]^{\mathcal{P}}$ (mixed attack policy)

$$\mathrm{P}_{R,M}(\text{capture}) = \sum_{P\in\mathcal{P}} m_P \,\mathrm{row}[P]\, x_P$$

diagonal matrix with all the elements of R

row vector with all the pure policies p_ℓ

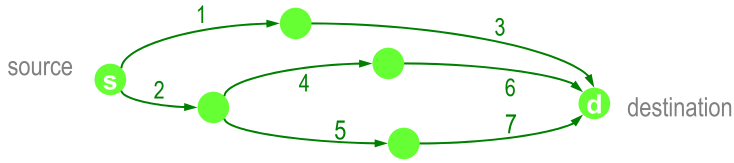
unique solution to

$$x_P = \mathrm{diag}[R]A(I - \mathrm{diag}[P])x_P + \mathrm{diag}[R]c$$

(matrix A and vector c only depend on the graph)

Linear (thus concave) in M (maximizer)
but not convex with respect to the routing policy R (minimizer)
so mini-max existence theorems do not apply…

Probability of capture



Under mild assumptions (*) on pure attacks

Given $R \in \mathcal{R}_{\text{no-cycle}}$, $M := \{ m_P : P \in \mathcal{P} \} \in [0,1]^{\mathcal{P}}$

$$\mathrm{P}_{R,M}(\text{capture}) = \Big(\sum_{P \in \mathcal{P}} m_P \,\mathrm{row}[P] \Big) x$$

row vector with all the pure policies p_ℓ

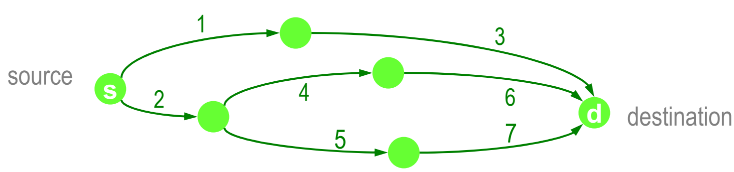
flow vector $\equiv$ unique solution to

$$x = \mathrm{diag}[R] A x + \mathrm{diag}[R] c$$

(matrix A and vector c only depend on the graph)

(*) the same pure attack does not simultaneously targets two links in the same path (true for every single-link or single-node attacks)

Probability of capture



Under mild assumptions (*) on pure attacks

Given $R \in \mathcal{R}_{\text{no-cycle}}$, $M := \{ m_P : P \in \mathcal{P} \} \in [0,1]^{\mathcal{P}}$

$$\mathrm{P}_{R,M}(\text{capture}) = \Big(\sum_{P \in \mathcal{P}} m_P \,\mathrm{row}[P] \Big) x$$

row vector with all the pure policies p_ℓ

flow vector $\equiv$ unique solution to

$$x = \mathrm{diag}[R] A x + \mathrm{diag}[R] c$$

(matrix A and vector c only depend on the graph)

Not convex with respect to the routing policy R but linear (convex!) with respect to the vector x…
Key idea: solve game for x & then compute R

Routing policies & Flow vectors

Theorem: i) There is a one-to-one correspondence between routing policies R in $\mathcal{R}_{\text{stoch}}$ & flow vectors x in a convex set $\mathcal{X} \subset \mathbb{R}^{\mathcal{L}}$

ii) For cycle-free $R \in \mathcal{R}_{\text{no-cycle}}$, the corresponding flow vector x satisfies

$$x = \text{diag}[R]Ax + \text{diag}[R]c$$

Therefore

$$\text{P}_{R,M}(\text{capture}) = \sum_{P \in \mathcal{P}} m_P \,\text{row}[P]x$$

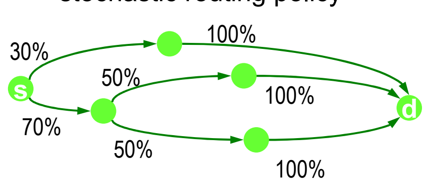
Routing policies & Flow vectors

Theorem: i) There is a one-to-one correspondence between routing policies R in $\mathcal{R}_{\text{stoch}}$ & flow vectors x in a convex set $\mathcal{X} \subset \mathbb{R}^{\mathcal{L}}$

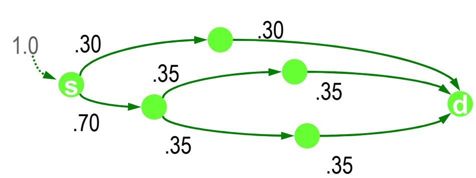
ii) For cycle-free $R \in \mathcal{R}_{\text{no-cycle}}$, the corresponding flow vector x satisfies

$$x = \text{diag}[R]Ax + \text{diag}[R]c$$

stochastic routing policy



flow vector



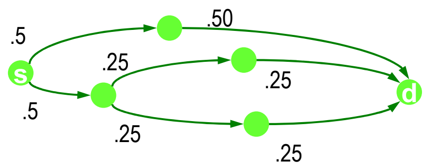
stochastic routing policy
$R := \{ .3, .7, 1, .5, .5, 1, 1\}$

flow vector
$x := \{ .3, .7, .3, .35, .35, .35, .35\}$

the vectors $x \in \mathcal{X}$ obey a "flow conservation law" at every node, with total unit flow exiting the source node

Flow game

flow vector



Compute saddle-point:

$x^* \in \mathcal{X}$ (flow vector)

$M^* \in [0,1]^{\mathcal{P}}$ (mixed attack policy)

for which

$$\sum_{P\in\mathcal{P}} m_P^* \operatorname{row}[P]x^* = \min_{x\in\mathcal{X}} \max_{M\in[0,1]^{\mathcal{P}}} \sum_{P\in\mathcal{P}} m_P \operatorname{row}[P]x$$

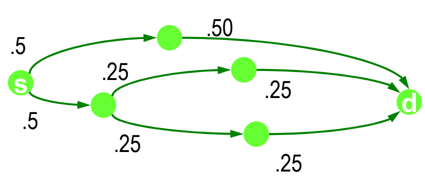
$$= \max_{M\in[0,1]^{\mathcal{P}}} \min_{x\in\mathcal{X}} \sum_{P\in\mathcal{P}} m_P \operatorname{row}[P]x.$$

Theorem: Every flow game has a saddle-point (x^*, M^*) with x^* cycle-free

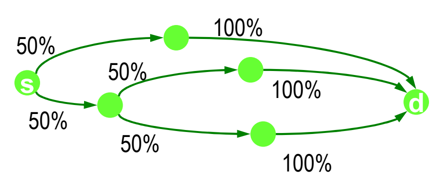
by bilinearity of the criterion and
convexity and (almost) compactness of $\mathcal{X}$ & $[0,1]^{\mathcal{P}}$

Back to routing game…

flow vector



stochastic routing policy



Theorem: The routing game has saddle-point policies.

Moreover, for every saddle-point (x^*, M^*) of the flow game with x^* cycle-free, the pair (R^*, M^*) is a saddle-point of the routing game, with R^* constructed from x^*:

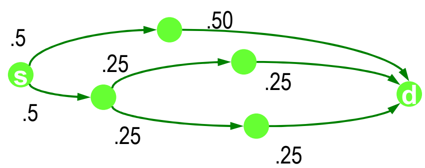
$$r_\ell^* := \frac{x_\ell^*}{\sum_{\ell'\in\mathcal{L}[\ell]} x_{\ell'}^*} \qquad \forall \ell \in \mathcal{L}$$

summation over all links that exit
from the same node as ℓ

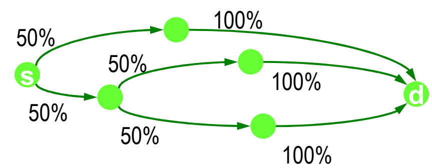
Solving the flow game actually solves the routing game…

Solution to the flow & routing games

flow vector



stochastic routing policy



Theorem: The value V^* of the flow game is given by

$$V^* = \min_{\substack{x \in \mathcal{X} \\ \mathrm{row}[P]x \le \mu,\ \forall P}} \mu$$

max-flow problem solvable by linear programming

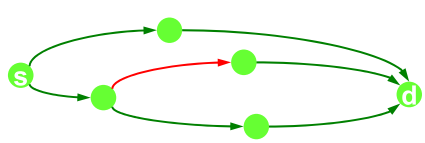
and the saddle-point x^* is any x at which the minimum is attained.

Optimal routing policy R^* can be computed using:

$$r^*_\ell := \frac{x^*_\ell}{\sum_{\ell' \in \mathcal{L}[\ell]} x^*_{\ell'}} \qquad \forall \ell \in \mathcal{L}$$

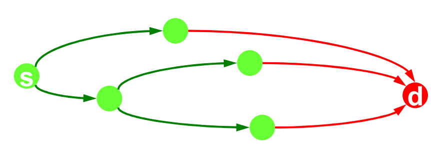
Max-flow interpretations

for pure attacks at individual links



- Optimal routing *minimizes the maximum link flow* (subject to constraints that depend on the link reliability)
- In practice, maximizes throughput subject to link bandwidth constraints

for pure attacks at individual nodes



- Optimal routing *minimizes the maximum node load* (subject to constraints that depend on node reliability)
- In practice, balances the load between nodes (useful for energy-starved nodes)

Several reasons to use multi-path routing

increase security

- Hespanha, Bohacek. Preliminary Results in Routing Games, 2001.
- Bohacek, Hespanha, Lee, Obraczka, Lim, Enhancing security via stochastic routing, 2002
- Papadimitratos, Haas, Secure message transmission in mobile ad hoc networks, 2003
- Lee, Misra, Rubenstein, Distributed Algorithms for Secure Multipath Routing, 2005

improve robustness

- Ganesan, Govindan, Shenker, Estrin, Highly Resilient, Energy Efficient Multipath Routing in Wireless Sensor Networks, 2002
- Wei, Zakhor, Robust Multipath Source Routing Protocol (RMPSR) for Video Communication over Wireless Ad Hoc Networks, 2004
- Tang, McKinley, A distributed multipath computation framework for overlay network applications, 2004

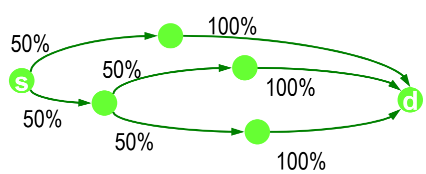
increase throughput

- Chen, Chan, Li, Multipath routing for video delivery over bandwidth-limited networks, 2004

maximize network utilization

- Elwalid, Jin, Low, Widjaja, MATE: MPLS adaptive traffic engineering, 2001
- Lee, Gerla, Split multipath routing with maximally disjoint paths in ad hoc networks, 2001
- Mirrokni, Thottan, Uzunalioglu, Paul, Simple polynomial time frameworks for reduced-path decomposition in multi-path routing, 2004

Latency-aware stochastic routing



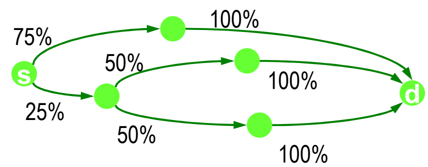
So far...

$$\text{Cost} = \text{P(capture)}$$

probability that packet will be captured

cost does not favor shorter paths

Latency-aware stochastic routing



So far...

$$\text{Cost} = \sum_{t=1}^{\infty} \text{P}(\text{capture at } t \text{ hop})$$

probability that packet will be captured

cost does not favor shorter paths…

One can bias routing towards shorter paths:

$$\text{Cost} = \sum_{t=1}^{\infty} (1+\epsilon)^{t-1}\, \text{P}(\text{capture at } t \text{ hop}) \qquad \epsilon > 0$$

more hops result in larger costs since $(1+\epsilon)^{t-1}$ increases with t

larger ϵ leads to larger penalty incurred by extra hop

Latency-aware routing game

$$\text{Cost} = \sum_{t=1}^{\infty} (1+\epsilon)^{t-1}\, \text{P}(\text{capture at } t \text{ hop}) \qquad \epsilon > 0$$

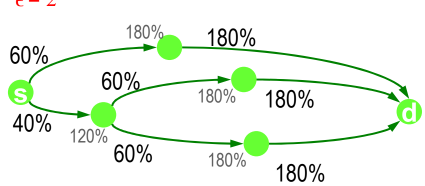
Theorem The latency-aware routing game has saddle-point policies.

These can be computed from the saddle-points of a flow game, where flow gets amplified by $(1+\epsilon)$ at every link.

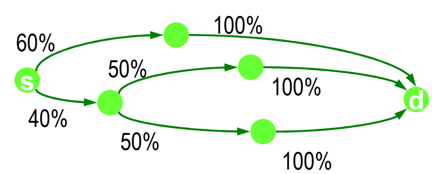
still solvable by linear programming

$\epsilon = 2$

flow vector



stochastic routing policy

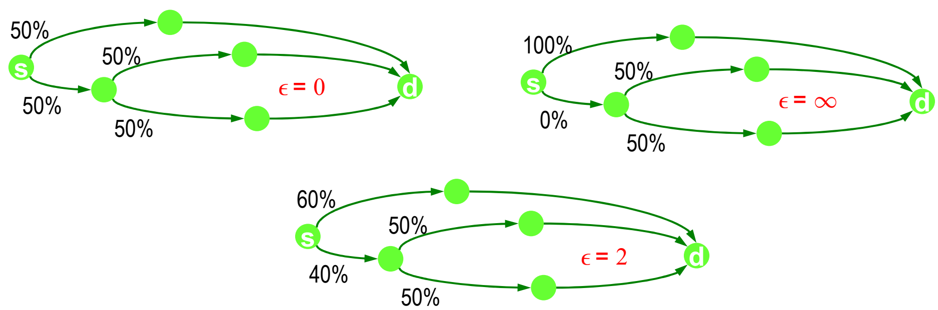


Latency-aware routing game

$$\text{Cost} = \sum_{t=1}^{\infty} (1+\epsilon)^{t-1}\, \mathrm{P}(\text{capture at } t \text{ hop}) \qquad \epsilon > 0$$

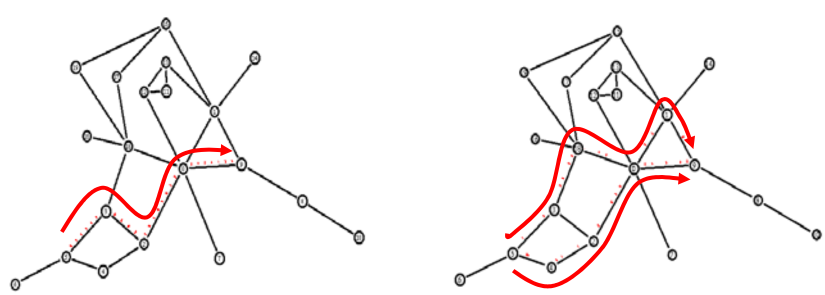
Theorem The latency-aware routing game has saddle-point policies.

These can be computed from the saddle-points of a flow game, where flow gets amplified by $(1+\epsilon)$ at every node.



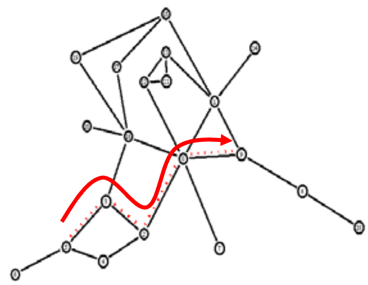
As $\epsilon \to \infty$, saddle-point converges to minimum hop routing
As $\epsilon \to 0$, saddle-point converges to max-flow routing

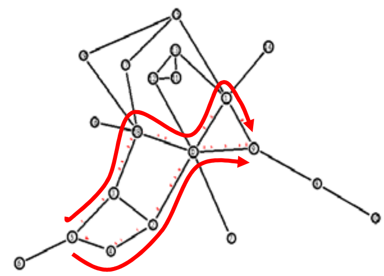
Outline



1. How to compute stochastic multi-path routing tables?
   Noncooperative game—explore redundancy in an adversarial context
   - Different levels of security/reliability for distinct links/nodes
   - Reduce latency
2. Other issues:
   - Scalable computation for large networks
   - Multi-path routing within the protocol stack

Computational complexity





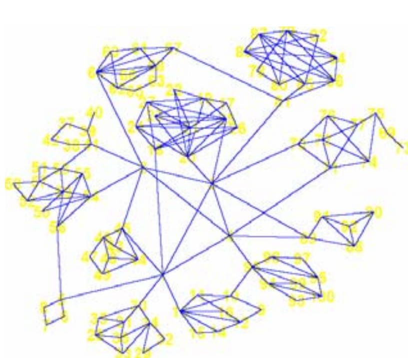
$n \equiv$ # of nodes

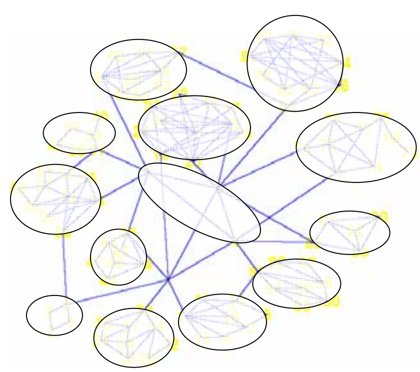
$n^2 \equiv$ # source/destination pairs

$O(n^5) \equiv$ max-flow routing total computation
(Goldberg & Tarjan pre-push algorithm, assuming dense worst-case dense graph. For Internet-like graphs more like $O(n^4 \log n)$)

How to improve this for large networks?

Hierarchical computation





1. Overall network graph is partitioned into several clusters of nodes (metanodes)
2. Routing is computed in two steps:

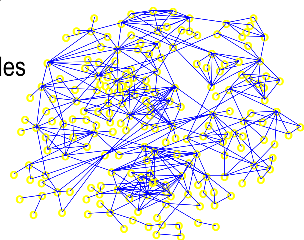
Option I: first inter- and then intra-metanode
S. Bohacek, JH, C. Lim, K. Obraczka. Hierarchical Max-Flow Routing. Feb. 2005. Submitted.

Option II: first intra- and then inter-metanode
J. Riehl, JH. Fractal Graph Optimization Algorithms. Mar. 2005. Submitted.

Either option reduces computation at the expense of a getting a suboptimal solution…

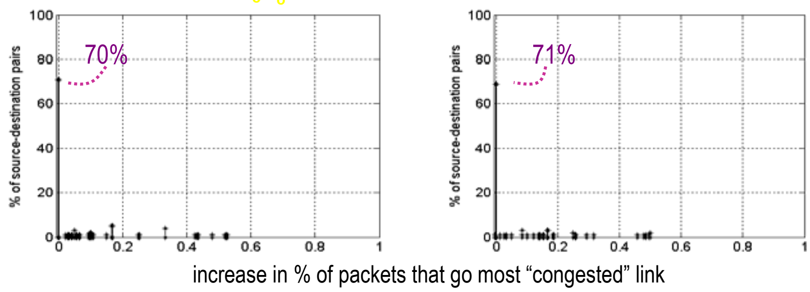
Flat vs. hierarchical (option I)

Exodus ISP
200 nodes
18 metanodes



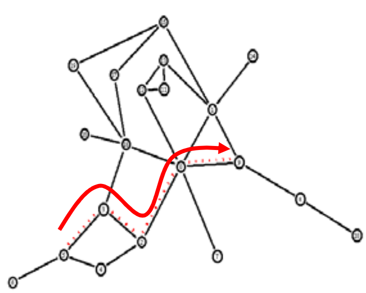
Verios ISP
960 nodes
41 metanodes

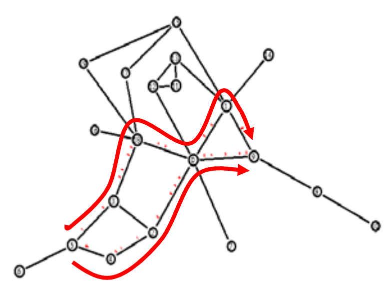
(topologies from Rocketfuel dataset)



routing computation down to $O(n^{3.2})$
over 70% of the source/destination pairs see no difference in the maximum number of packets that go through any link (measure of fragility)

Outline





1. How to compute multi-path routing tables?
   - Explores redundancy in an adversarial context
   - Different levels of security/reliability for distinct links/nodes
2. Other issues:
   - Scalable computation for larger networks
   - Multi-path routing within the protocol stack

Protocol stack

UCSB

TCP/IP protocol stack:

increased abstraction

Application layer	user interface
Transport layer	congestion control and error recovery (TCP or UDP)
Network layer	routing
Medium access layer	mediates access to shared medium
Physical layer	interface with hardware

Key principles behind layered architectures:

1. Each layer provides a specific service to the layers above
2. The implementation details of one layer should be "transparent" to all layers above, as long as it provides its designated service.

Multi-path routing & the protocol stack

UCSB

TCP/IP protocol stack:

increased abstraction

Application layer	user application
Transport layer	congestion control and reliability (UDP or TCP)
Network layer	routing
Medium access layer	mediates access to shared medium
Physical layer	interface with hardware

Transport layer

Assumes

network layer forwards packets from source to destination, possibly losing and/or re-ordering some packets

Guarantees

UDP – nothing!

TCP – network will not be overloaded with traffic (congestion control), all packets eventually arrive at dest. in the correct order (reliability)

Multi-path routing & the protocol stack

UCSB

However, most versions of TCP (NewReno, SACK) also assumes that…

1. re-ordering never exceeds two packets

pkt 1	pkt 3	pkt 4	pkt 2	pkt 5	pkt 6	… okay
pkt 1	pkt 3	pkt 4	pkt 5	pkt 2	pkt 6	… assumed lost

2. all loses are due to congestion and therefore trigger a reduction in the sending rate

∴ persistent re-ordering leads to very small sending rates

Assumes

network layer forwards packets from source to destination, possibly losing and/or re-ordering some packets

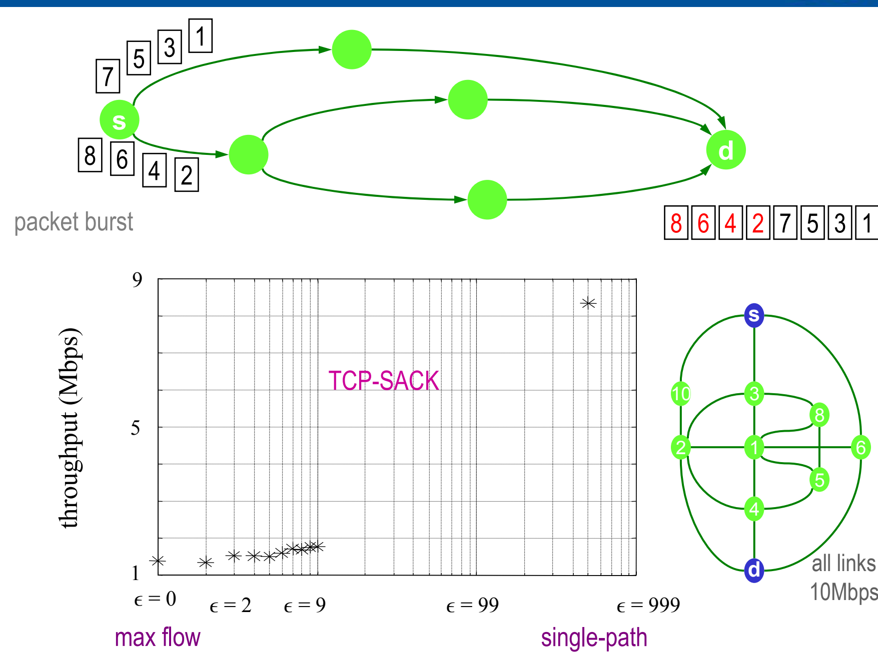
Guarantees

UDP – nothing!

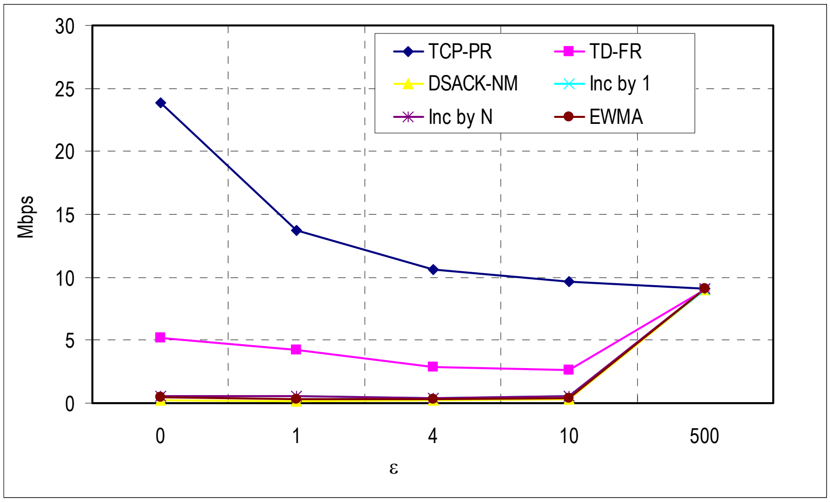
TCP – network will not be overloaded with traffic (congestion control), all packets eventually arrive at dest. in the correct order (reliability)

Packet re-ordering in multi-path routing

UCSB



TCP versions for persistent re-ordering



TCP-PR [Bohacek,JH,Lee,Lim,Obraczka 03]
TD-FR [Paxson 97]
remaining (DSACK option) [Blanton,Allman 02]

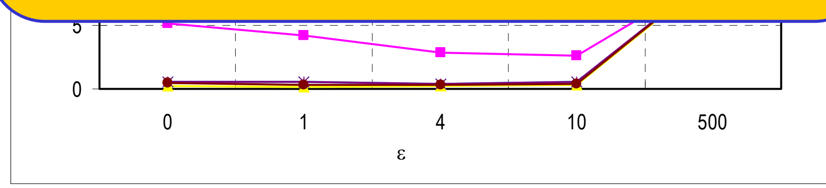
ordering not used at all to detect drops
does not use the DSACK option so
only needs changes in server side

TCP versions for persistent re-ordering

However…

For real-time applications (such as control) one generally does not want to rely on a "general-purpose" reliable transport protocol like TCP
(retransmits old data until arrival can be confirmed)

Instead, one probably wants to use UDP and develop protocols (at the application layer) that are more adequate for networked control systems…



TCP-PR [Bohacek,JH,Lee,Lim,Obraczka 03]
TD-FR [Paxson 97]
remaining (DSACK option) [Blanton,Allman 02]

ordering not used at all to detect drops
does use DSACK option
only needs changes in server side

Conclusions

- Communication networks are extremely vulnerable components to critical systems
  - multitude of individual components, spatially distributed, difficult to protect
  - especially true for wireless networks (jamming, eavesdropping, battery drainage due to overuse, etc.)
- Game theory is a natural framework for network security
  - redundancy, by itself, will not solve the problem
  - attacks are not random events: very unlikely events can be prompted by an attacker
- Determined routing polices that are robust with respect to attacks
  - formulation as a zero-sum game between router and attacker
  - saddle-point solutions found by reducing problem to a flow-game (solvable by linear programming)
  - policies found also have applications to throughput maximization and load balancing
  - other formulations are possible (Markov games, leading to distance vector algorithms)
- Other challenges
  - scalability (addressable by hierarchical or distributed computation)
  - compatibility with transport layer (use newest TCP versions or UDP)

References

Stochastic routing

- S. Bohacek, J. Hespanha, K. Obraczka. Saddle Policies for Secure Routing in Communication Networks. In *Proc. 41st CDC*, Dec. 2002.
- S. Bohacek, J. Hespanha, J. Lee, C. Lim, K. Obraczka. Enhancing security via stochastic routing. In *Proc. of the 11th IEEE ICCCN*, May 2002.
- J. Hespanha, S. Bohacek. Preliminary Results in Routing Games. In *Proc. ACC*, June 2001.

TCP under persistent packet re-ordering

- S. Bohacek, J. Hespanha, J. Lee, C. Lim, K. Obraczka. A New TCP for Persistent Packet Reordering-TCP-PR. Oct. 2003. Submitted to the *IEEE/ACM Trans. on Networking*.
- S. Bohacek, J. Hespanha, J. Lee, C. Lim, K. Obraczka. TCP-PR: TCP for Persistent Packet Reordering. In *Proc. of the IEEE ICDCS*, May 2003.

Hierarchical routing

- S. Bohacek, J. Hespanha, C. Lim, K. Obraczka. Hierarchical Max-Flow Routing. Feb. 2005. Submitted.
- J. Riehl, J. Hespanha. Fractal Graph Optimization Algorithms. Mar. 2005. Submitted.

http://www.ece.ucsb.edu/~hespanha