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

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

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

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

Attack 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...
1. How to compute stochastic multi-path routing tables for general networks?
   Noncooperative game—explore redundancy in an adversarial context

2. Multi-path routing for multi-agent & networked control systems

Outline

Stochastic routing policies

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

for every node $n$

$\sum_{\ell \in \mathcal{L}[n]} r_\ell = 1$

set of (unidirectional) links

source

destination

probability that a packet arriving at the node where $\ell$ starts will be routed though link $\ell$

summation over links that exit node $n$

add to 1

add to 1

set of all routing policies

R_{stoch} \equiv \text{set of all routing policies}
Stochastic routing policies

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

$\mathcal{R}_\text{stoch} \equiv \{ r_\ell \geq 0 : \ell \in \mathcal{L} \}$

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

Attack space

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

- pure attack at link 3 with 10% probability of success:
  \[ P_3 := \{ 0, 0, 1, 0, 0, 0, 0 \} \]
- pure attack at node 6 with 20% probability of success:
  \[ P_6 := \{ 0, 0, .2, 0, 0, .2, .2 \} \]
Mixed attacks

probability that the attacker will intercept a packet traveling in link $\ell$.

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

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

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

Example

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

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

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

stochastic routing policy \( R \equiv \{ .3, .7, 1, .5, .5, 1, 1 \} \)

pure attacks available to attacker \( P \equiv \{ \{1,0,0,0,0,0,0\},\{0,1,0,0,0,0,0\}, \ldots \} \)

mixed attack policy \( M \equiv \{ 0, 0, .33, .33, .33, 0, 0 \} \)

\[ P_{R,M}(\text{capture}) = (30\% \times 33\% + 2 \times 35\% \times 33\%) \times 10\% = 3.3\% \]

but not really rational...

Example

mixed attack policy \( M \equiv \{ 0, 0, .33, .33, .33, 0, 0 \} \)

stochastic routing policy \( R \equiv \{ .3, .7, 1, .5, .5, 1, 1 \} \)

\[ P_{R,M}(\text{capture}) = (30\% \times 33\% + 2 \times 35\% \times 33\%) \times 10\% = 3.3\% \]

for this attack policy \( M \)

but attacker could do better against \( R \) with

\[ P_{R,M}(\text{capture}) = (70\% \times 100\%) \times 10\% = 7\% \]

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)
- $M^* \in [0,1]^p$ (mixed attack policy)

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

$$P_{R^*,M^*}(\text{capture}) = \min_{R \in \mathcal{R}_{\text{no-cycle}}} \max_{M \in [0,1]^p} P_{R,M}(\text{capture}) = \max_{M \in [0,1]^p} \min_{R \in \mathcal{R}_{\text{no-cycle}}} P_{R,M}(\text{capture})$$

Existence? Computation?

probability of capture

Given

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

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

row vector with all the pure policies $p_i$

unique solution to

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

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

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

(row vector with all the pure policies \( p_i \))

flow vector \( x = \text{diag}[R]Ax + \text{diag}[R]c \)

(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)

---

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 \( R \)
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 \( \mathcal{X} \subset \mathbb{R}^L \)

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

\[
x = \text{diag}[R]A x + \text{diag}[R]e
\]

Therefore

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

Routing policies & Flow vectors

\begin{itemize}
  \item stochastic routing policy
  \[ R := \{.3, .7, 1, .5, .5, 1, 1\} \]
  \item flow vector
  \[ x := \{.3, .7, .3, .35, .35, .35, .35\} \]
\end{itemize}

the vectors \( x \in \mathcal{X} \) obey a “flow conservation law” at every node, with total unit flow exiting the source node.
**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 \(X \times [0,1]^P\).

Compute saddle-point:

\[
\begin{align*}
    x^* &\in X \quad \text{(flow vector)} \\
    M^* &\in [0,1]^P \quad \text{(mixed attack policy)}
\end{align*}
\]

for which

\[
\begin{align*}
    \sum_{P \in P} m_P^* \row{P}{x^*} &= \min_{x \in X} \max_{M \in [0,1]^P} \sum_{P \in P} m_P \row{P}{x} \\
    &= \max_{M \in [0,1]^P} \min_{x \in X} \sum_{P \in P} m_P \row{P}{x}.
\end{align*}
\]

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 \(X \times [0,1]^P\).

**Back to routing game...**

**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 \(x^*:\)

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

summation over all links that exit from the same node as \(\ell\).

Solving the flow game actually solves the routing game...
Theorem: The value $V^*$ of the flow game is given by

$$V^* = \min_{x \in \mathcal{X}} \mu \quad \text{max-flow problem solvable by linear programming}$$

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

Optimal routing policy $R^*$ can be computed using:

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

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**
- 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

---

Estimation through network

**process**

\[ x_{k+1} = Ax_k + Bw_k \]

**zero-mean stochastic disturbance**

\[ x_{k+1} \]

**remote state-estimator**

\[ \hat{x}_{k+1} = A\hat{x}_k \]

for simplicity:
- full-state available
- no measurement noise
- no quantization

Optimal remote state estimator:

\[
\hat{x}_{k+1} = \begin{cases} 
Ax_k & \text{succ. transmission at time } k \\
A\hat{x}_k & \text{unsucc. transmission at time } k 
\end{cases}
\]

Remote state estimation error:

\[ e_k := x_k - \hat{x}_k \]

\[ e_{k+1} = \begin{cases} 
Bw_k & \text{succ. transmission at time } k \\
Ae_k + Bw_k & \text{unsucc. transmission at time } k 
\end{cases} \]
Estimation through network

\[ x_{k+1} = Ax_k + Bw_k \]

Failures caused by an attacker that succeeds with probability \( p_{\text{att}} \)

- single-path routing \( \equiv \) probability of failed transmission = \( p_{\text{att}} \)
  \[ e_{k+1} = \begin{cases} Bw_k & \text{w.p. } 1 - p_{\text{att}} \\ A \epsilon_k + Bw_k & \text{w.p. } p_{\text{att}} \end{cases} \]
  mean-square stable iff \( p_{\text{att}} < \frac{1}{\lambda(A)} \)

- multi-path routing \( \equiv \) probability of failed transmissions = \( p_{\text{att}}/2 \)
  \[ e_{k+1} = \begin{cases} Bw_k & \text{w.p. } 1 - \frac{p_{\text{att}}}{2} \\ A \epsilon_k + Bw_k & \text{w.p. } \frac{p_{\text{att}}}{2} \end{cases} \]
  mean-square stable iff \( p_{\text{att}} < \frac{2}{\lambda(A)} \)

Consider random failures:

- \( p_{\text{fail}} \equiv \) probability that a link will fail
- \( T_{\text{tr}} \equiv \) mean time-to-recover (exponentially distributed)

At steady state:

\[ \text{P(unsucc. transmission)} = \frac{p_{\text{fail}}T_{\text{tr}}}{1 + p_{\text{fail}}T_{\text{tr}}} \]

but drops are not i.i.d. …
Estimation through network

\[ x_{k+1} = Ax_k + Bw_k \]

Consider random failures:

- Probability that a link will fail: \( p_{\text{fail}} \)
- Mean time-to-recover: \( T_{\text{ttr}} \) (exponentially distributed)

For a 1-dimensional quasi-stable process \( A = 1 + \varepsilon, \varepsilon \ll 1 \):
- Low fail probability: \( p_{\text{fail}} \ll 1 \)
- Single-path routing: mean-square stable if \( T_{\text{ttr}} \leq \frac{1}{2\varepsilon} \)
- Multi-path routing: mean-square stable if \( T_{\text{ttr}} \leq \frac{1}{\varepsilon} \) twice as large admissible mean time-to-recover

in this networked estimation problem, the maximum spread of packets is optimal even “against” random failures

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 to study robustness
  - Redundancy, by itself, does not guarantee robustness
  - Attacks are not random events: very unlikely events can be prompted by an attacker
- Determined routing policies that exploit multi-path routing
  - Formulation as a zero-sum game between router and attacker
  - Saddle-point solutions found by reducing problem to a flow-game
  - Policies found also have applications to
    - Throughput maximization
    - Load balancing
    - Improve robustness of NCSs (even against random failures)

[ Observation: traditional measures of QoS such as probability of drop, expected delay are not sufficient to predict performance in NCSs ]