Modeling, Analysis, and Mitigation of Internet Worm Attacks

Presenter: **Cliff C. Zou**

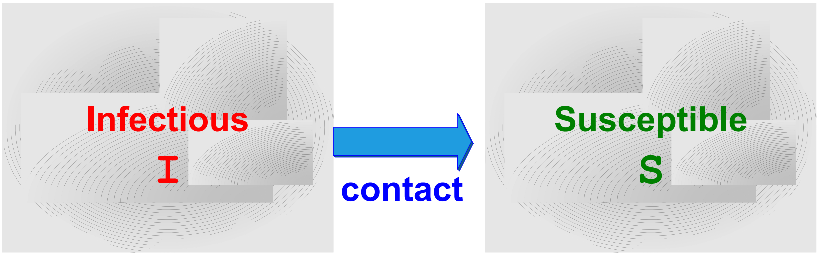
Dept. of Electrical & Computer Engineering
University of Massachusetts, Amherst
Advisor: Weibo Gong, Don Towsley
Joint work with Don Towsley, Weibo Gong, Lixin Gao, and Songlin Cai

UMassAmherst

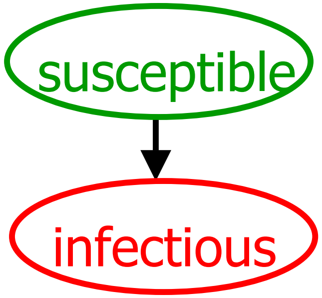
Outline

- **Introduction of epidemic models**
- Two-factor worm model
- Early detection and monitoring
- Feedback dynamic quarantine defense
- Routing worm: a fast, selective attack worm
- Worm scanning strategies
- Summary and future work

Epidemic Model —
Simple Epidemic Model



# of contacts $\propto$ **I** × **S**



$S(t)$: # of susceptible

N : # of hosts

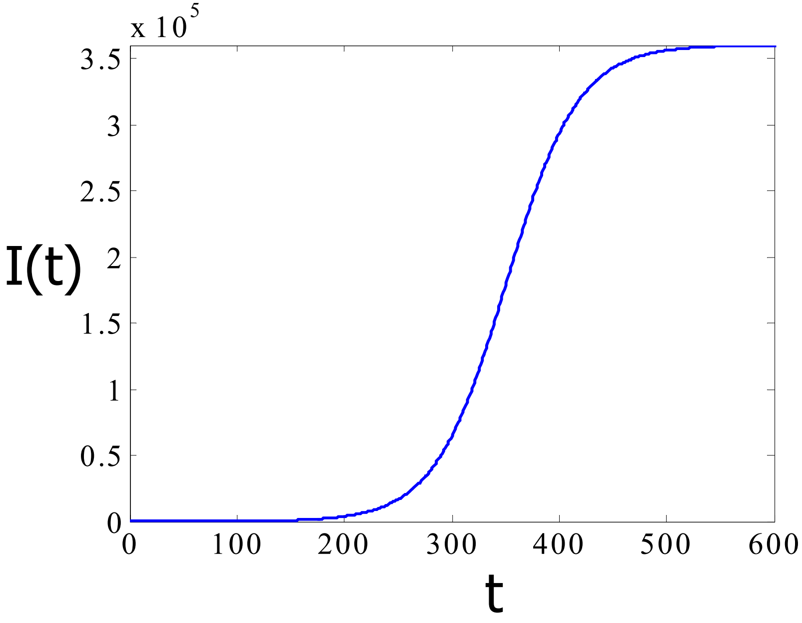
$I(t)$: # of infectious

β : infection ability

Simple epidemic model for fixed population homogeneous system:

$$\dot{I}(t) = \beta I(t) \cdot S(t)$$

$$N = I(t) + S(t)$$



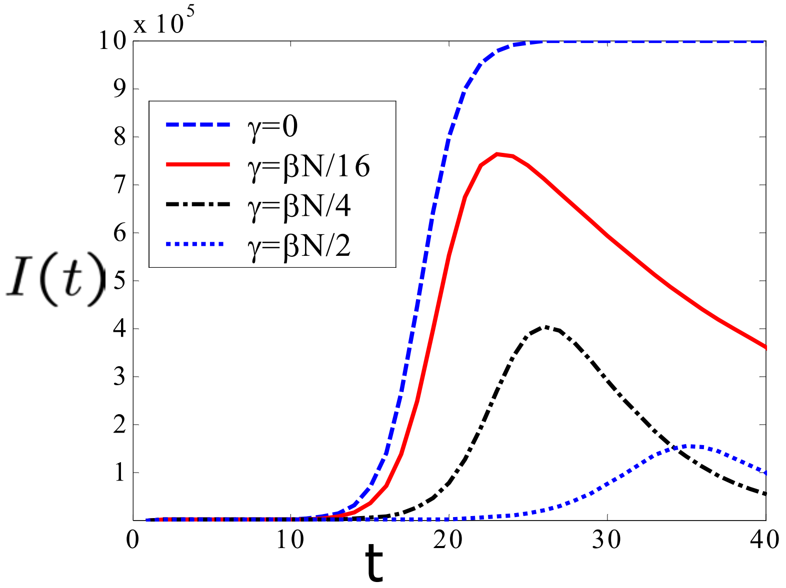
Epidemic Model — Kermack-McKendrick Model

- State transition: susceptible → infectious → removed

$U(t)$: # of removed from infectious γ : removal rate

$$\dot{I}(t) = \beta I(t)S(t) - \dot{U}(t)$$

$$\dot{U}(t) = \gamma I(t) \qquad S(t) + I(t) + U(t) = N$$



- Epidemic threshold theorem:
  - No *outbreak* happens if

$$S(0) < \rho \qquad (\dot{I}(t) < 0,\ \forall t > 0)$$

where $\rho \equiv \gamma / \beta$

ρ : epidemic threshold

Outline

- Introduction of epidemic models
- **Two-factor worm model**
- Early detection and monitoring
- Feedback dynamic quarantine defense
- Routing worm: a fast, selective attack worm
- Worm scanning strategies
- Summary and future work

Internet Worm Modeling — Consider Human Countermeasures

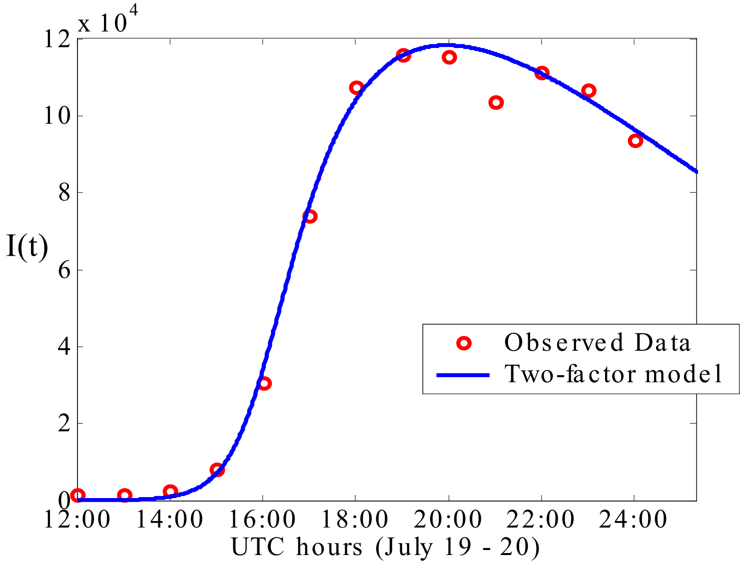
- Human countermeasures:
  - Clean and patch: download cleaning program, patches.
  - Filter: put filters on firewalls, gateways.
  - Disconnect computers.
- Reasons for:
  - Suppress most new viruses/worms from outbreak.
  - Eliminate virulent viruses/worms eventually.
- Removal of both susceptible and infectious hosts.

Internet Worm Modeling —
Two-Factor Worm Model

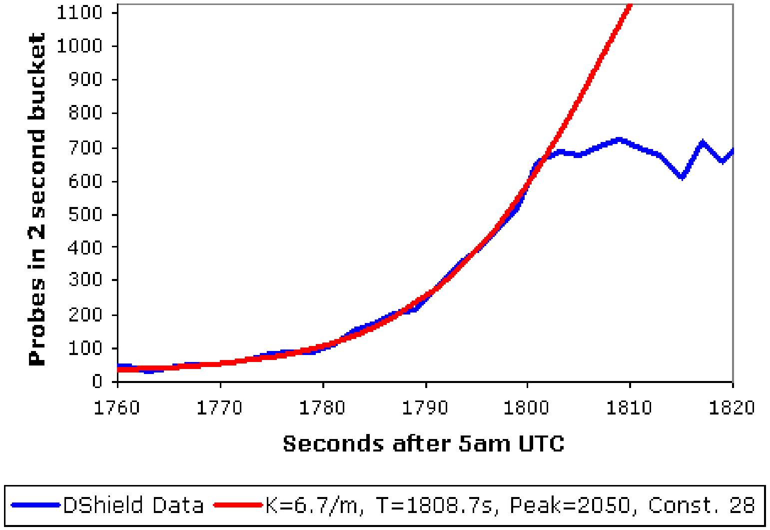
- Factor #2: Network congestion
  - Large amount of scan traffic.
  - Most scan packets with unused IP addresses (30% BGP routable)
  - Effect: slowing down of worm infection ability $\beta \Rightarrow \beta(t)$
- *Two-factor worm* model (extended from KM model):
  - $\beta(t)$: Slowed down infection ability due to congestion
  - $V(t)$: removal from susceptible hosts. $U(t)$:from infectious

$$\begin{cases} \dot{S}(t) & = -\beta(t)S(t)I(t) - \dot{V}(t) \\ \dot{V}(t) & = \mu S(t)[I(t) + U(t)] \\ \dot{U}(t) & = \gamma I(t) \\ N & = S(t) + I(t) + U(t) + V(t) \end{cases}$$

Verification of the *Two-Factor Worm Model*



Code Red



SQL Slammer *

- Conclusion:
  - Simple epidemic model overestimates a worm's propagation
  - At beginning, we can ignore these two factors.

* Figure from:

D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, N. Weaver,
"Inside the Slammer Worm", *IEEE Security & Privacy*, July 2003.

Summary of Two-Factor Model

- Modeling Principle:
  - We must consider the changing environment when we model a dynamic process.
- Two factors affecting worm propagation:
  - Human countermeasures.
  - Worm's impact on Internet infrastructure.
- At the early stage of worm propagation, we can ignore these two factors.
  - Still use simple epidemic model.

Outline

- Introduction of epidemic models
- Two-factor worm model
- **Early detection and monitoring**
- Feedback dynamic quarantine defense
- Routing worm: a fast, selective attack worm
- Worm scanning strategies
- Summary and future work

How to detect an unknown worm at its early stage?

- Monitoring:
  - Monitor worm scan traffic (non-legitimate traffic).
    - Connections to nonexistent IP addresses.
    - Connections to unused ports.
  - Observation data is very **noisy**.
    - Old worms' scans.
    - Port scans by hacking toolkits.
- Detecting:
  - Anomaly detection for unknown worms
  - Traditional anomaly detection: threshold-based
    - Check traffic burst (short-term or long-term).
    - Difficulties: **False alarms; threshold tuning**.

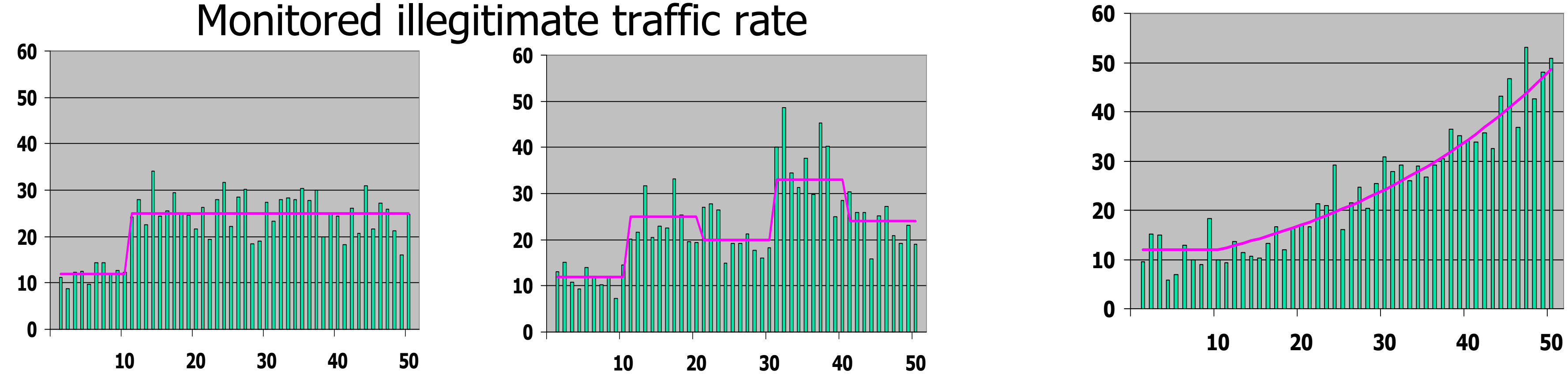
“Trend Detection”

— Detect traffic *trend*, not *burst*

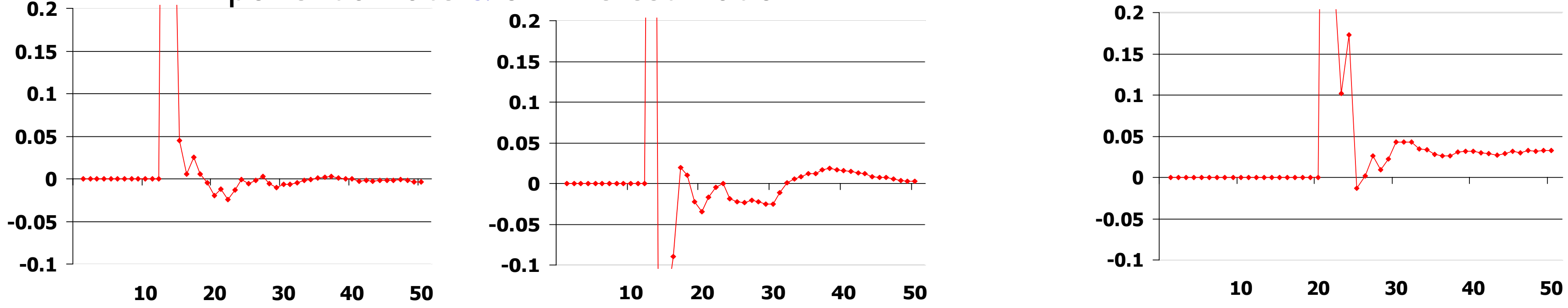
Trend: worm exponential growth trend at the beginning $\dot{I}(t) = \alpha I(t)$

Detection: the exponential rate should be a positive, constant value

Monitored illegitimate traffic rate



Exponential rate α on-line estimation



Non-worm traffic burst

Worm traffic

Why exponential growth at the beginning?

- The law of natural growth — reproduction
- Exponential growth – fastest growth pattern when:
  - Negligible interference (beginning phase).
  - All objects have similar reproductive capability.
  - Large-scale system – law of large number.
- Fast worm has exponential growth pattern
  - Attacker's incentive: infect as many as possible before people's counteractions.
  - If not, a worm does not reach its spreading speed limit.
  - Slow spreading worms can be detected by other ways.

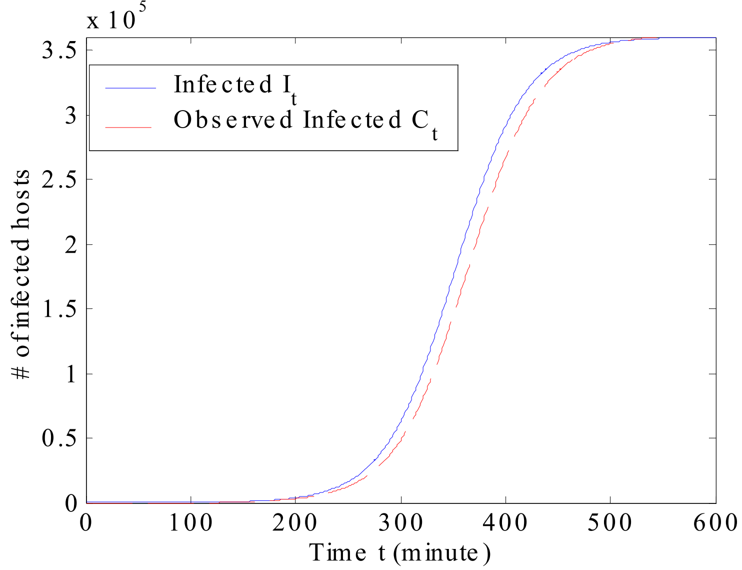
Code Red simulation experiments

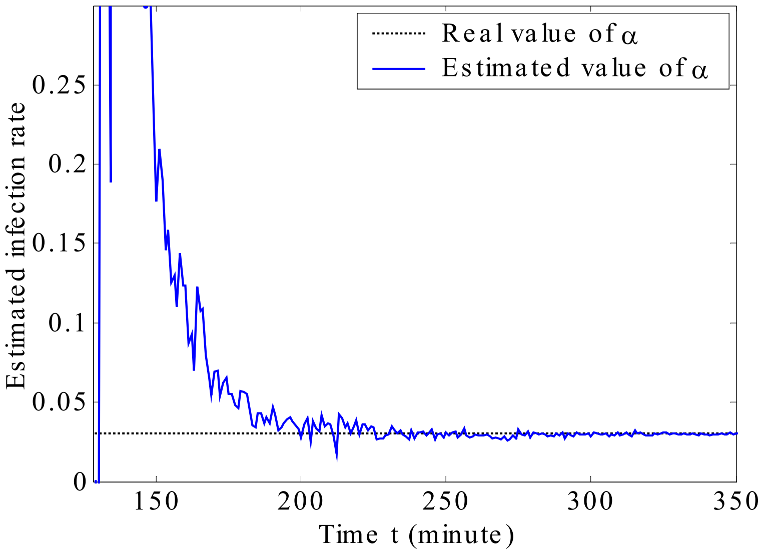
Population: N=360,000, Infection rate: α = 1.8/hour,

Scan rate η = N(358/min, 100^2), Initially infected: I_0=10

Monitored IP space 2^{20}, Monitoring interval: Δ = 1 minute

Consider background noise

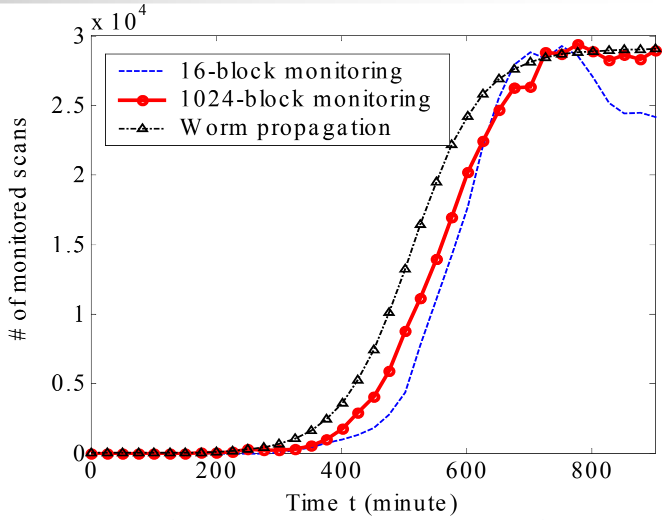




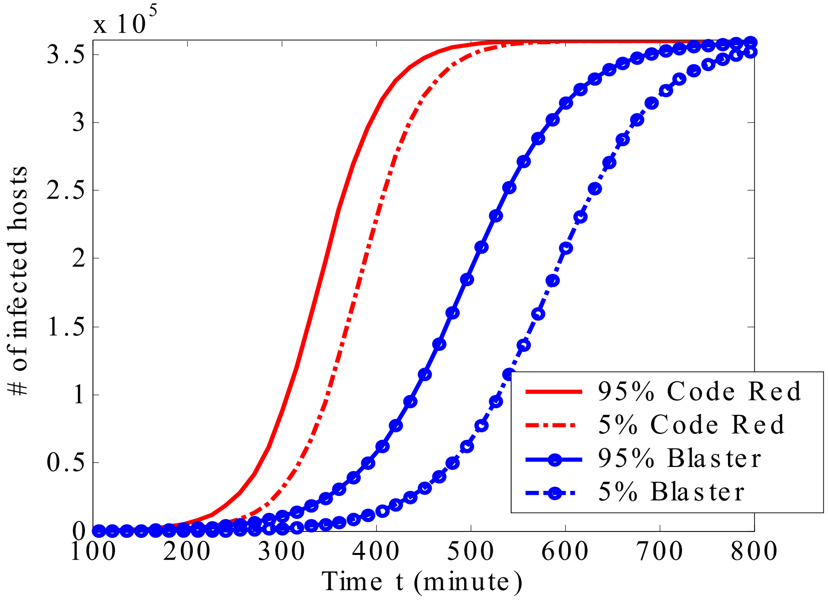
Before 2% (223 min): estimate is already stabilized and oscillating a little around a positive constant value

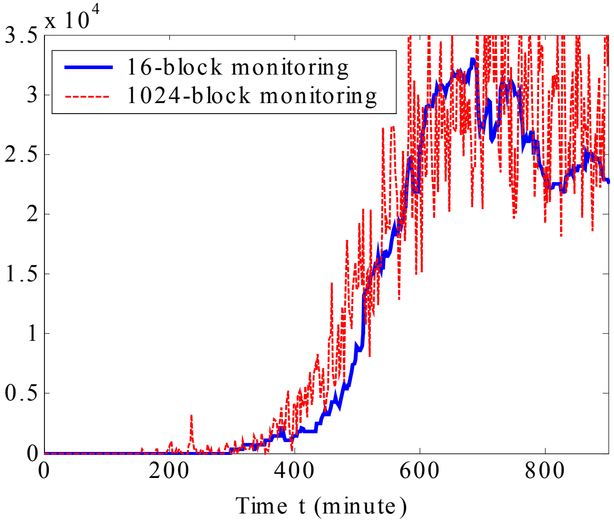
Early detection of Blaster

- Blaster: sequentially scans from a starting IP address:
  - 40% from local Class C address.
  - 60% from a random IP address.
- It follows simple epidemic model.



After using low-pass filter





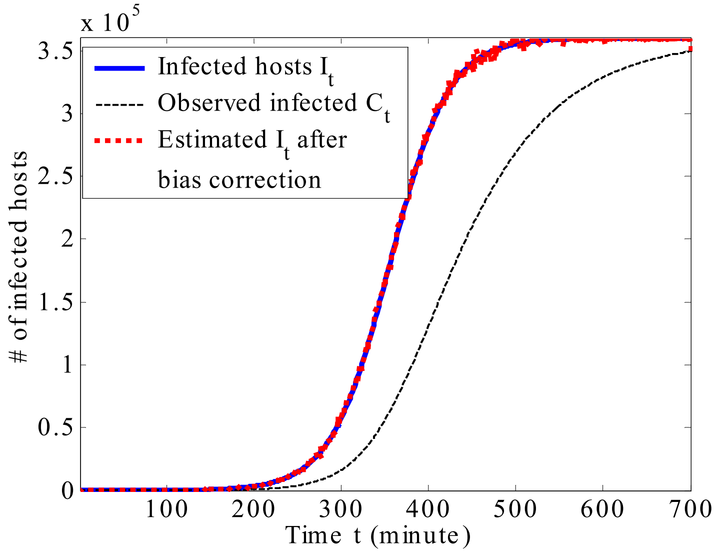
Bias correction for uniform-scan worms

- *Bernoulli trial* for a worm to hit monitors (hitting prob. = p).

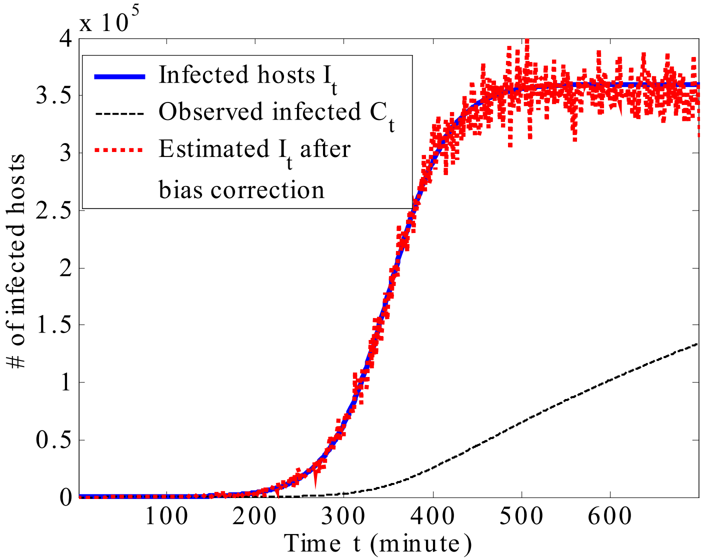
Bias correction:

$$\hat{I}_k = \frac{C_{k+1} - (1-p)^\eta C_k}{1-(1-p)^\eta}$$

η : Average scan rate



Monitoring 2^{17} IP space



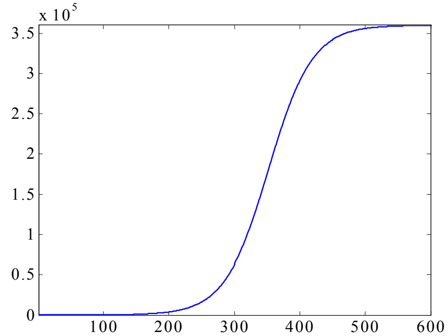
Monitoring 2^{14} IP space

Bias correction can provide unbiased estimate of I(t)

Prediction of Vulnerable population size N

Direct from Kalman filter: $X_t = [\; 1 + \alpha\Delta \quad \beta \;]$

$$\alpha = \beta N \quad \rightarrow \quad \hat{N} = \frac{\hat{\alpha}}{\hat{\beta}}$$

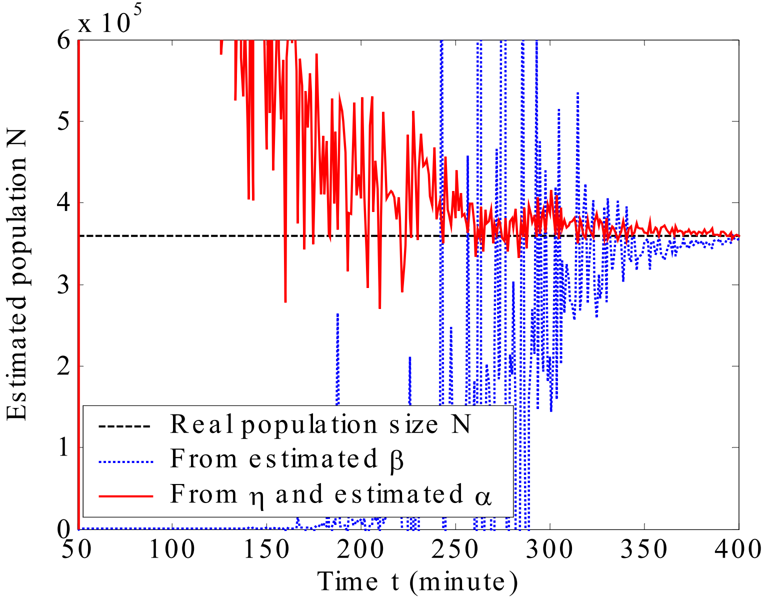


Alternative method:

η : A worm sends out η scans per Δ time

(derived from egress scan monitor)

$$\alpha = \eta N / 2^{32} \quad \rightarrow \quad \hat{N} = \frac{2^{32}\hat{\alpha}}{\eta}$$



Estimation of population N

Summary of Early Detection

- **Trend detection**: non-threshold based methodology
  - Principle: **detect traffic *trend*, not *burst***
  - Pros : Robust to background noise → low false alarm rate
- Monitoring requirement for non-uniform scan worm:
  - Monitor many well-distributed IP blocks; low-pass filter
- For uniform-scan worms
  - Bias correction: $\hat{I}_t = \frac{C_{t+1} - (1-p)^{\eta} C_t}{1-(1-p)^{\eta}}$
  - Forecasting N: $N = \alpha \cdot 2^{32} / \eta$ (IPv4)

$$\alpha = \beta N \quad \Rightarrow \quad \beta = \eta / \Omega \quad \Rightarrow$$ Routing worm

Ω : scanning IP space α : Infection rate η : Average scan rate

p : scan hitting prob. C_t : cumulative # of observed infectious

Outline

- Introduction of epidemic models
- Two-factor worm model
- Early detection and monitoring
- **Feedback dynamic quarantine defense**
- Routing worm: a fast, selective attack worm
- Worm scanning strategies
- Summary and future work

Motivation: automatic mitigation and its difficulties

- Fast spreading worms pose serious challenges:
  - SQL Slammer infected 90% within 10 minutes.
  - Manual counteractions out of the question.
- Difficulty of automatic mitigation — high false alarm cost.
  - Anomaly detection for unknown worm.
  - False alarms vs. detection speed.
  - Traditional mitigation:
    - No quarantine at all → … → long-time quarantine until passing human's inspection.

Principles in real-world epidemic disease control

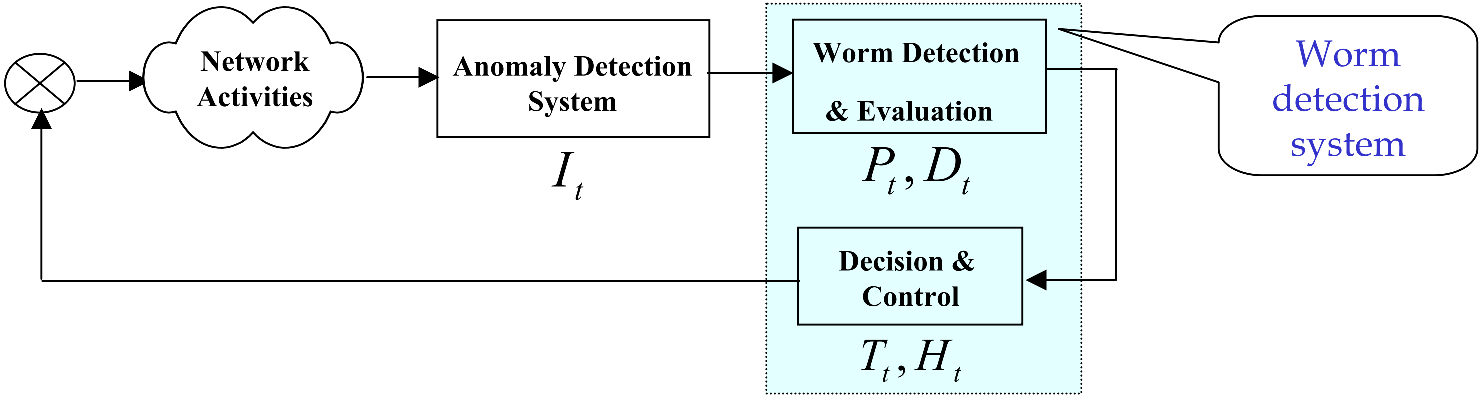
- Principle #1 — Preemptive quarantine
  - Assuming guilty before proven innocent
  - Comparing with disease ***potential*** damage, we are willing to pay for ***certain*** false alarm cost.
- Principle #2 — Feedback adjustment
  - More serious epidemic, more aggressive quarantine action
    - Adaptive adjustment of the trade-off between disease damage and false alarm cost.

Dynamic Quarantine

- Assuming guilty before proven innocent
  - Quarantine on suspicion, release quarantine after a short time ***automatically*** ← reduce false alarm cost
  - Can use any host-based, subnet-based (e.g., CounterMalice) anomaly detection system.
  - Host or subnet based quarantine (not whole network-level quarantine).
  - Quarantine is on suspicious port only.
- A *graceful* automatic mitigation:

No quarantine → Dynamic short-time quarantine → long-time quarantine

Feedback Control Dynamic Quarantine Framework (host-level)



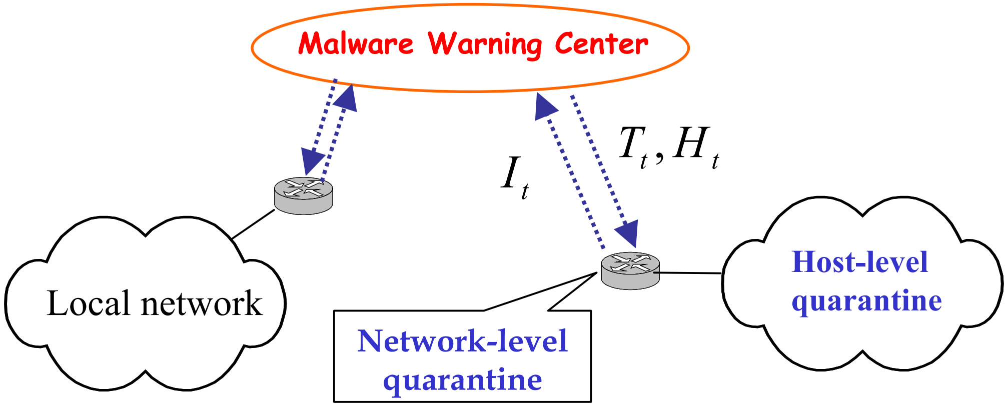
- Feedback : More suspicious, more aggressive action
- Predetermined constants: U, V (for each TCP/UDP port)
- Observation variables: I_t :# of quantined hosts/subnets.
- Worm detection and evaluation variables:

Probability	$P_t = f_1(I_t, V, U),$	$I_t \uparrow \rightarrow P_t \uparrow$
Damage	$D_t = f_2(I_t, \dot{I}_t, V, U),$	$I_t, \dot{I}_t \uparrow \rightarrow D_t \uparrow$

- Control variables:

Quarantine time	$T_t = g_1(P_t, D_t, I_t, V, U),$	$P_t, D_t \uparrow \rightarrow T_t \uparrow$
Alarm threshold	$H_t = g_2(P_t, D_t, I_t, V, U),$	$P_t, D_t \uparrow \rightarrow H_t \downarrow$

Two-level Feedback Control
Dynamic Quarantine Framework



- Network-level quarantine (Internet scale)
  - Dynamic quarantine is on routers/gateways of local networks.
  - Quarantine time, alarm threshold are recommended by MWC.
- Host-level quarantine (local network scale)
  - Dynamic quarantine is on individual host or subnet in a network.
  - Quarantine time, alarm threshold are determined by:
    - Local network’s worm detection system.
    - Advisory from Malware Warning Center.

Host-level Dynamic Quarantine without Feedback Control

- First step: no feedback control/optimization
  - Fixed quarantine time, alarm threshold. $T_t,\ H_t$

I(t): # of infectious S(t): # of susceptible **T**: Quarantine time

R(t): # of quarantined infectious **Q(t)**: # of quarantined susceptible

λ_1: quarantine rate of infectious λ_2: quarantine rate of susceptible

$$R(t) = \int_{t-T}^{t} [I(\tau) - R(\tau)]\lambda_1 d\tau \;\; - \int_{t-T}^{t} \gamma R(\tau) d\tau$$

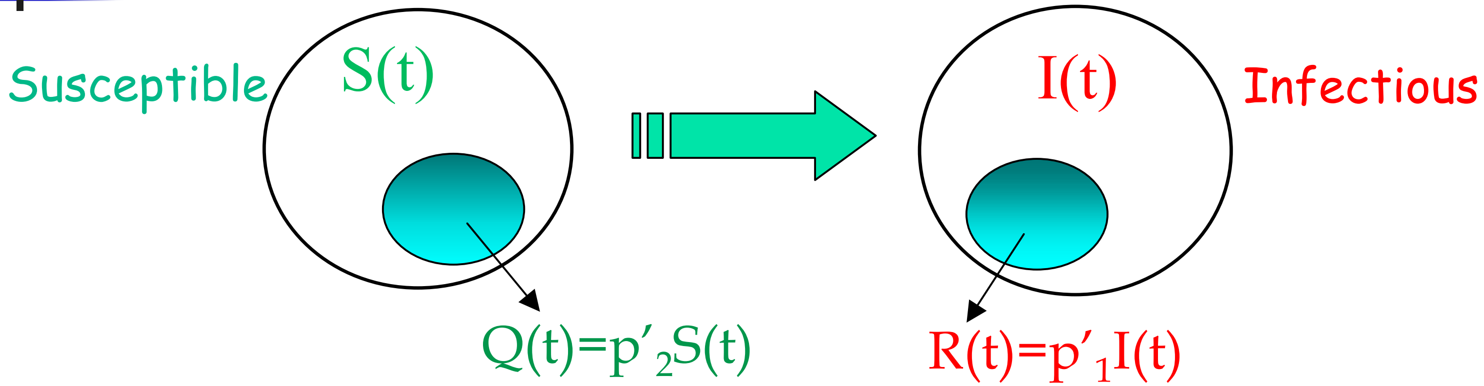
removed

Assumptions: $$\begin{cases} R(\tau) & \simeq R(t) \\ I(\tau) & \simeq I(t) \end{cases} \quad \forall \tau \in [t-T, t]$$

$$\Rightarrow \qquad R(t) = [I(t) - R(t)]\lambda_1 T$$

$$\Rightarrow \qquad R(t) = p'_1 I(t) \qquad p'_1 = \frac{\lambda_1 T}{1+\lambda_1 T}$$

Extended Simple Epidemic Model



Susceptible $S(t)$

$I(t)$ Infectious

$Q(t)=p'_2S(t)$

$R(t)=p'_1I(t)$

$$\text{\# of contacts} \propto [S(t)-Q(t)]\times[I(t)-R(t)]$$

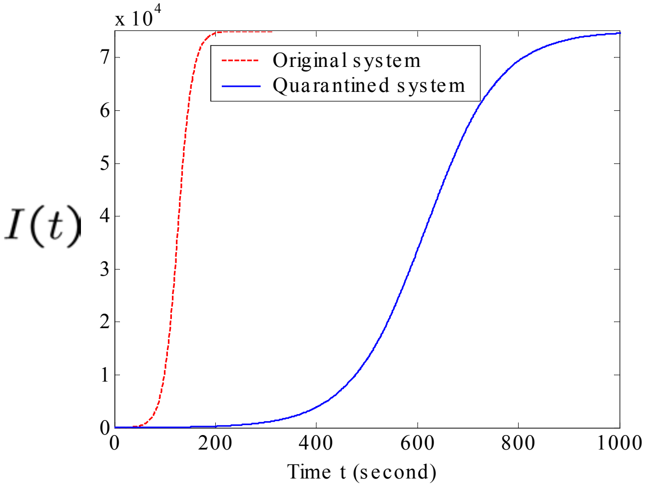
Before quarantine: $\dot{I}(t)=\beta I(t)\cdot S(t)$

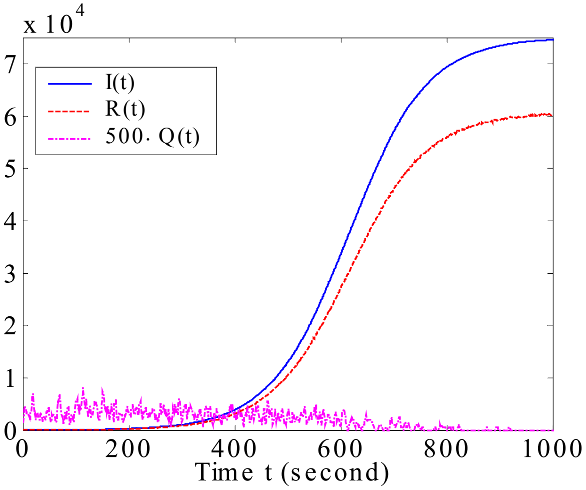
After quarantine:

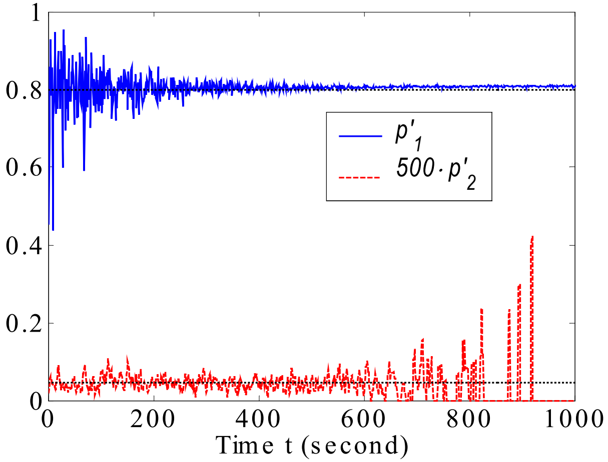
$$\begin{aligned}\dot{I}(t) &= \beta[I(t)-R(t)][S(t)-Q(t)]\\ &= \beta' I(t)\cdot S(t)\end{aligned}$$

$$\beta' = (1-p'_1)(1-p'_2)\beta$$

Extended Simple Epidemic Model







Vulnerable population N=75,000, worm scan rate 4000/sec
T=4 seconds, $\lambda_1 = 1$, $\lambda_2=0.000023$ (twice false alarms per day per node)

R(t): # of quarantined infectious $R(t) = p'_1 I(t)$ $p'_1 = \frac{\lambda_1 T}{1+\lambda_1 T}$

Q(t): # of quarantined susceptible $Q(t) = p'_2 S(t)$ $p'_2 = \frac{\lambda_2 T}{1+\lambda_2 T}$

$$R(t) = \int_{t-T}^{t} [I(\tau) - R(\tau)]\lambda_1 d\tau$$
$$Q(t) = \int_{t-T}^{t} [S(\tau) - Q(\tau)]\lambda_2 d\tau$$

$\Leftarrow$ Law of large number

Summary of Feedback Dynamic Quarantine Defense

- Learn the quarantine principles in real-world epidemic disease control:
  - **Preemptive quarantine**: Comparing with disease ***potential*** damage, we are willing to pay ***certain*** false alarm cost
  - **Feedback adjustment**: More serious epidemic, more aggressive quarantine action
- Two-level feedback control dynamic quarantine framework
  - Optimal control objective:
    - Reduce worm spreading speed, # of infected hosts.
    - Reduce false alarm cost.
- Derive worm models under open-loop dynamic quarantine
  - Efficiently reduce worm spreading speed
  - Raise/generate epidemic threshold

Outline

- Introduction of epidemic models
- Two-factor worm model
- Early detection and monitoring
- Feedback dynamic quarantine defense
- **Routing worm: a fast, selective attack worm**
- Worm scanning strategies
- Summary and future work

BGP Routing Worm

- Contains BGP routing prefixes:
  - Fact: routable IP space < 30% of entire IPv4 space.
- Scanning space is 28.6% of entire IPv4 space.
  - Increasing worm's speed by 3.5 times (Sept. 22, 2003).
- Payload requirement: 175KB
  - Non-overlapping prefixes:
    - Remove "128.119.85/24" if BGP contains "128.119/16".
  - 140602 prefixes → 62053 prefixes (Sept. 22, 2003)
  - Big payload for Internet-scale worm propagation.

Class A Routing Worm

- IANA provides Class A address allocations
  - Class A (x.0.0.0/8); 256 Class A in IPv4 space.

002/8 : IANA - Reserved
003/8 : General Electric Company
056/8 : U.S. Postal Service
214/8 : US-DOD
216/8 : ARIN
217/8 : RIPE NCC
224/8 : IANA - Multicast

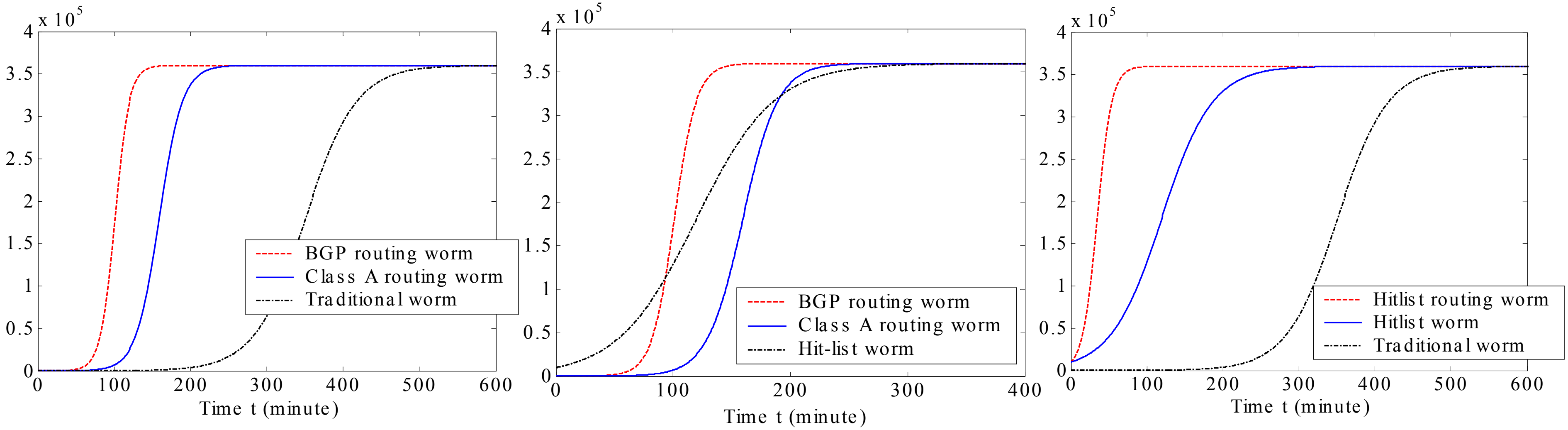
- 116 Class A networks contain all BGP routable space.
  - Scanning space: 45.3%; payload: 116 Bytes.
- Routing worm based on BGP prefixes aggregation.
  - Trade-off: scanning space ↔ Prefix payload ("/13" ⇒ 37%, 5KB)

Routing Worm Propagation Study

$$\dot{I}(t) = \beta I(t)[N - I(t)]$$ where $$\beta = \frac{\eta}{\Omega}$$

N : # of vulnerable η : Scan rate Ω : Scanning space

Comparison of the Code Red worm, a routing worm, a hit-list worm, and a hit-list routing worm



N=360,000; η=358 scans/min; I(0)=10 (10,000 for the hit-list worm)

Routing Worm: A Selective Attack Worm

- Selective Attack
  - Different behaviors on different compromised hosts.
  - Imposes damage based on geographical information of IP addresses of compromised hosts
- Geographical information of IP addresses
  - IP address → Routing prefix → AS ⇐ BGP routing table

    AS → Company, ISP, Country ⇐ Researches
  - *Pinpoint* attacking vulnerable hosts in a specific target
  - Potential terrorists cyberspace attacks

Selective Attack: a Generic Attacking Technique

- Imposes damage based on *any* information a worm can get from compromised hosts
  - OS (e.g. : illegal OS, OS language, time zone)
  - Software (e.g. : installed a specific program)
  - Hardware (e.g. : CPU, memory, network card)
- Improving propagation speed
  - Maximize usage of each compromised host.
    - Multi-thread worm: generates different numbers of threads based on CPU, memory, and connection speed of compromised computers.

Defense: Upgrading IPv4 to IPv6

- Routing worm idea: **Reducing worm scanning space**
  - Effective, easier than hit-list worm to implement
  - Difficult to prevent:
    - public BGP tables and IP geographical information
- Defense: **Increasing worm scanning space**

  — Upgrading IPv4 to IPv6
  - The smallest network in IPv6 has 2^{64} IP address space.
  - A worm needs *40 years* to infect 50% of vulnerable hosts in a network when N=1,000,000, η=100,000/sec, I(0)=1000
  - Limitation: for scan-based worms only

Summary of Routing Worm

- Routing worm: a worm containing information of BGP routing prefixes in the worm code.
- Routing worm: a faster spreading worm
  - Scans routable space (< 30%) instead of entire IPv4 space.
  - Increasing propagation speed by 2 ~ 3.5 times.
- Routing worm: a selective attack worm
  - IP address → routing prefix → AS → ISP, Country
    - Pinpoint attacking vulnerable hosts in a specific target
  - Selective attack based on *any* information a worm can get from compromised hosts.
- Defense: Increase a worm's scanning space

⇒ IPv4 upgrade to IPv6

Outline

- Introduction of epidemic models
- Two-factor worm model
- Early detection and monitoring
- Feedback dynamic quarantine defense
- Routing worm: a fast, selective attack worm
- **Worm scanning strategies**
- Summary and future work

Epidemic Model Introduction

- Model for homogeneous system

$$\frac{dI(t)}{dt} = \beta I(t)[N - I(t)]$$

N : # of hosts

$I(t)$: # of infectious

β : infection ability

η : scan rate

Ω : scanning space

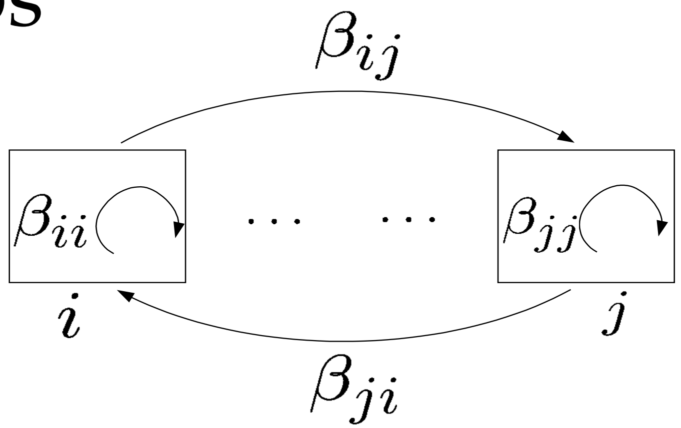
For worm modeling:

$$\beta = \eta/\Omega \quad \Leftarrow \text{Infinitesimal analysis}$$

- Model for interacting groups

$$\frac{dI_i(t)}{dt} = \beta_{ii} I_i(t)[N_i - I_i(t)] + \sum_{j \neq i} \beta_{ji} I_j(t)[N_i - I_i(t)]$$

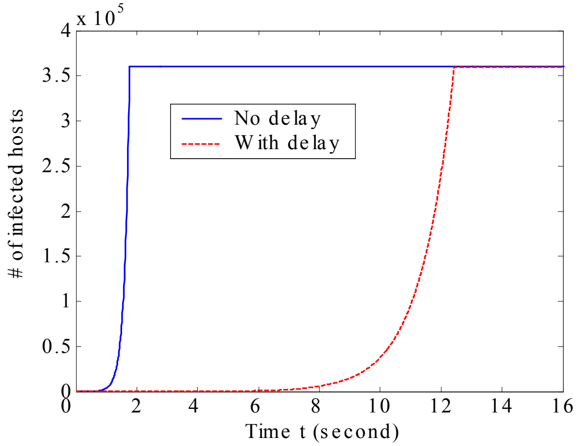
$$\text{for } i = 1, 2, \cdots, K$$



Idealized Worm

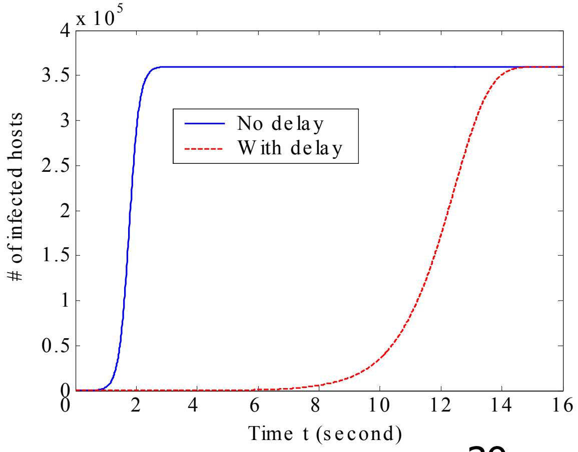
- Knows IP addresses of *all* vulnerable hosts
- Perfect worm
  - Cooperation among worm copies

$$\frac{dI(t)}{dt} = \begin{cases} \eta I(t-\epsilon), & I(t) < N \\ 0, & I(t) = N \end{cases}$$



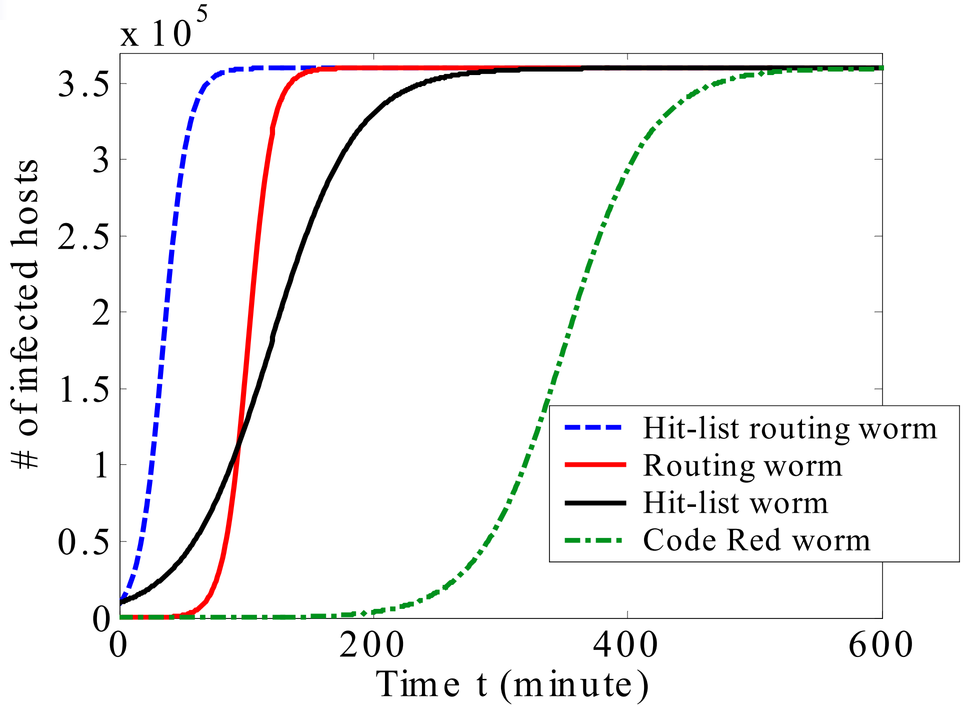
- Flash worm
  - No cooperation; random scan

$$\frac{dI(t)}{dt} = \frac{\eta}{N} I(t-\epsilon)[N - I(t)]$$



- Complete infection within seconds

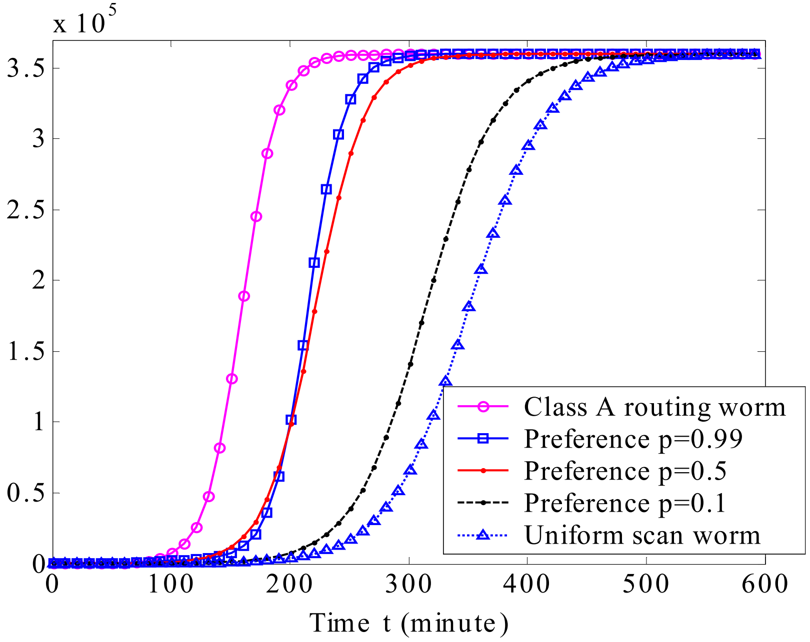
Uniform Scan Worms



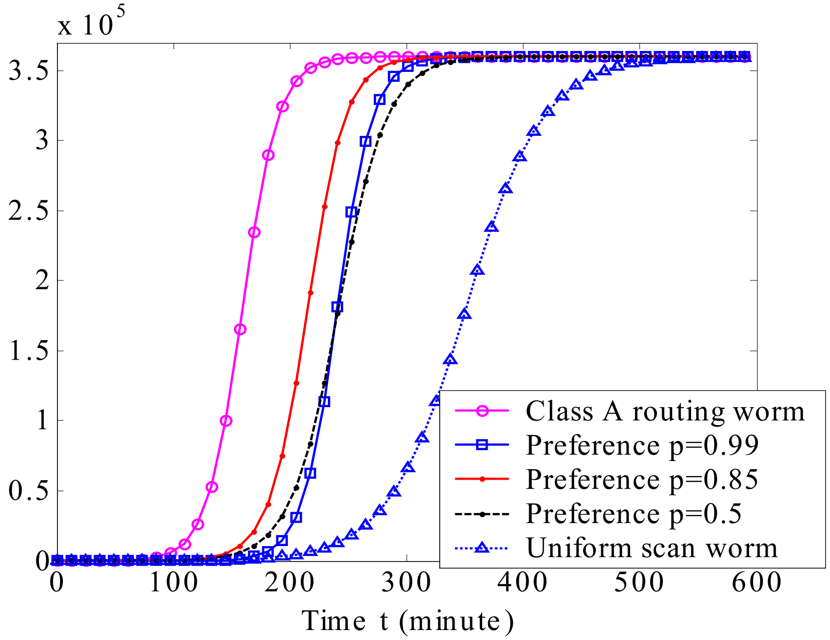
- Hit-list worm has a hit-list of I(0)=10,000
- Routing worm has $\Omega=0.286\times 2^{32}$
- Other parameters:
  N=360,000
  η=358/min
  I(0)=10

- Defense: Crucial to prevent attackers from
  - Identifying IP addresses of a large number of vulnerable hosts → Flash worm, Hit-list worm
  - Obtaining address information to reduce a worm's scanning space → Routing worm

Local Preference Scan Worm



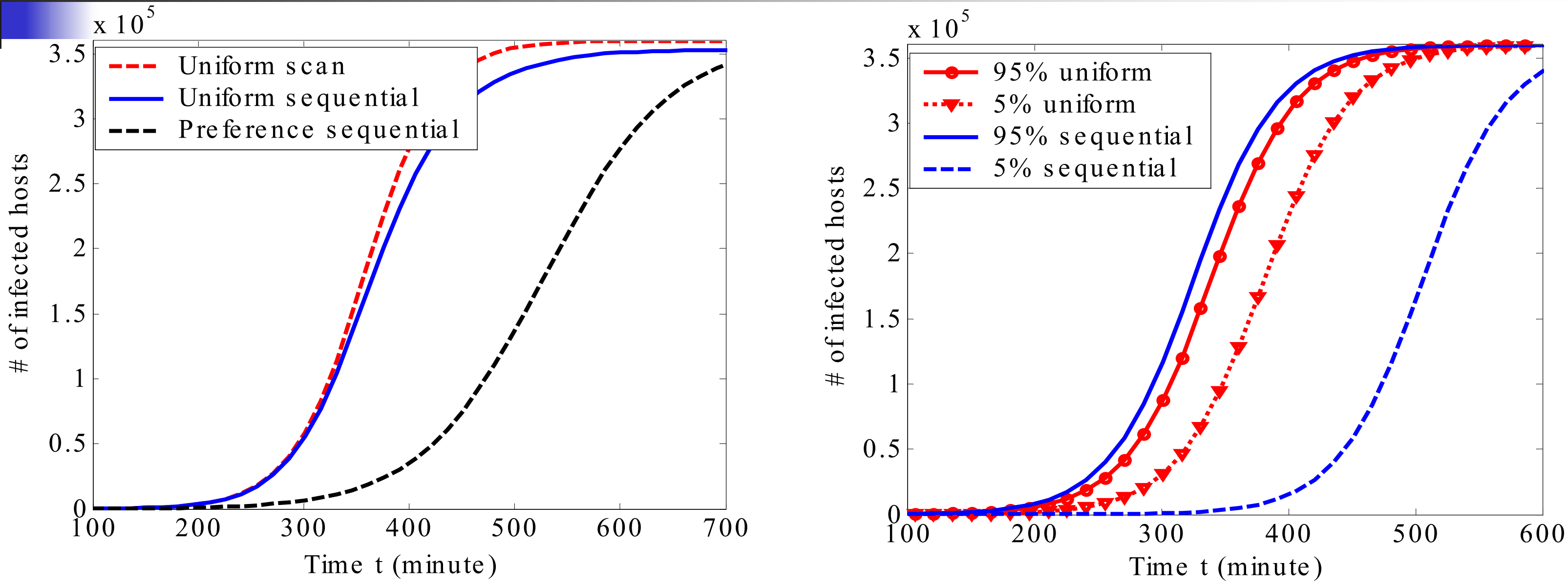
Class A local scan (K=256, m=116)



Class B local scan (K=2^{16}, m=116×2^8)

- Local preference scan increases speed (when vulnerable hosts are not uniformly distributed)
- Local scan on Class A ("/8") networks: $p^* \rightarrow 1$
- Local scan on Class B ("/16") networks: $p^* \cong 0.85$
- Code Red II: p=0.5 (Class A), p=0.375 (Class B) $\Leftarrow$ Smaller than p^*

Sequential Scan Worm Simulation Study



Uniform scan, sequential scan with/without local preference (100 simulation runs)

Vulnerable hosts uniformly distributed in BGP routable IP space (28.6% of IPv4 space)

- Local preference in selecting starting point is a bad idea.
- Sequential scan ≡ uniform scan
  (when vulnerable hosts are uniform distributed)
- *Mean value analysis* cannot analyze variability.

Summary of Worm Scanning Strategies

- Modeling basis:
  - Law of large number; mean value analysis; infinitesimal analysis.
  - Epidemic model: $\frac{dI(t)}{dt} = \frac{\eta}{\Omega} I(t)[N - I(t)]$
- Conclusions:
  - All about worm scanning space Ω (or density of vulnerable population):
    - Flash worm, Hit-list worm, Routing worm
    - Local preference, divide-and-conquer, selective attack

Outline

- Introduction of epidemic models
- Two-factor worm model
- Early detection and monitoring
- Feedback dynamic quarantine defense
- Routing worm: a fast, selective attack worm
- Worm scanning strategies
- **Summary and future work**

Worm Research Summary

- Modeling and analysis:
  - Two-factor worm model.
    - *Human counteractions* and *network congestion*.
  - Routing worm.
  - Worm scanning strategies.

$$\frac{dI(t)}{dt} = \frac{\eta}{\Omega} I(t)[N - I(t)]$$

- Worm defense:
  - Early detection: *detect trend, not burst*.
  - Feedback dynamic quarantine
    - *preemptive quarantine* and *feedback adjustment*.
- Papers at: http://tennis.ecs.umass.edu/~czou

Future Work

- Feedback dynamic quarantine defense.
  - Enterprise network.
  - Cost function; optimal control.
- Verification on real data.
  - Early detection.
  - Statistical analysis.
- Realistic Internet-scale worm simulation.
  - First: distribution of on-line hosts.