

# Analysis and Detection of Internet Worms

D. Towsley  
U. Massachusetts

Collaborators: W. Gong, C. Zou

# Motivation

- Code Red worm
  - more than 360,000 infected in less than one day.
  - random scan
- SQL Slammer
  - less than 15 minutes to infect 100,000 hosts
  - congested portions of Internet
- Blaster

Can we model behavior?

# Motivation

- Code Red worm
- SQL Slammer
- Blaster

## Can we model behavior?

- models can be used to:
  - understand worm spreading behavior
  - detect worm propagation and damage
  - develop effective mitigation techniques

# Motivation

- Code Red worm
- SQL Slammer
- Blaster

## Can we model behavior?

- models can be used to:
  - understand worm spreading behavior
  - detect worm propagation and damage
  - develop effective mitigation technique

# Worm spreading factors

## □ scan rate

- host speed, bandwidth
- protocol
- payload size

## □ scanning strategy

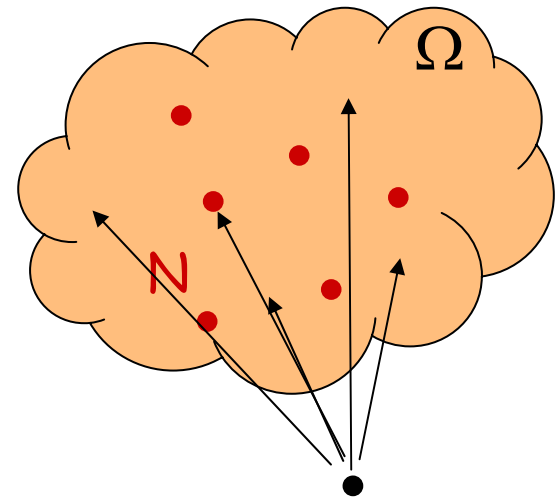
- uniform vs. local preference
- sequential vs. random
- optimizations
  - hit-list
  - BGP scan

# Outline

- worm spreading models
  - scanning strategies
- worm detection/estimation
- summary

# Worm spreading model

- address space, size  $\Omega$
- $N$  vulnerable hosts
- scan rate (per host),  $\eta$
- pairwise infection rate,  
 $\beta = \eta / \Omega$



# Simple worm spreading model

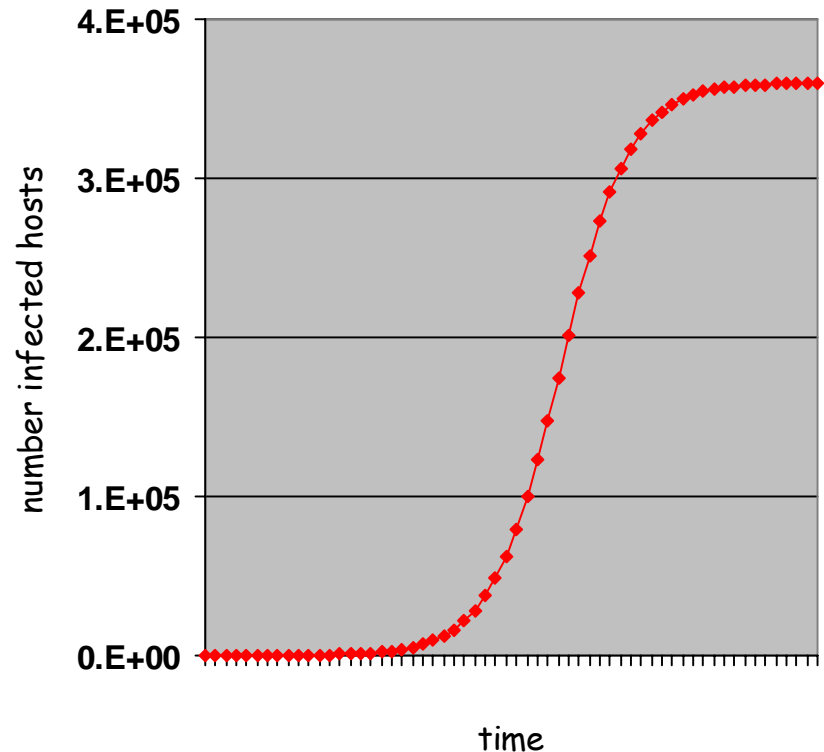
$I(t)$  - number of infected hosts

Epidemic model:

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

with infection rate

$$\begin{aligned} \alpha &= \beta N \\ &= \eta N/\Omega \end{aligned}$$



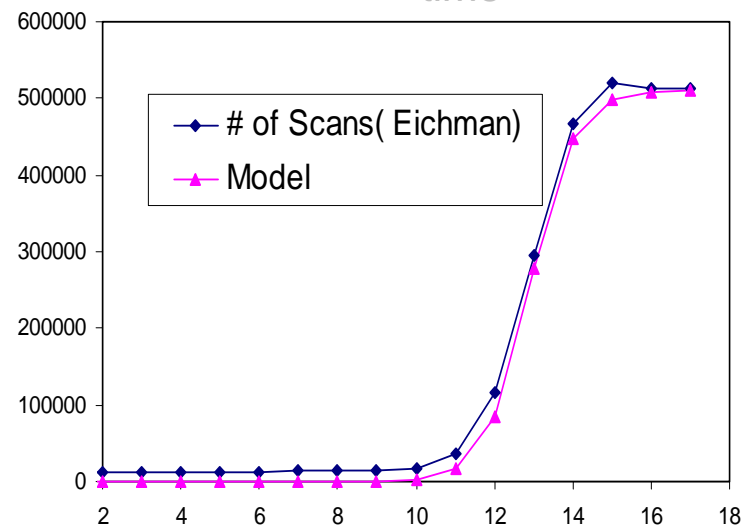
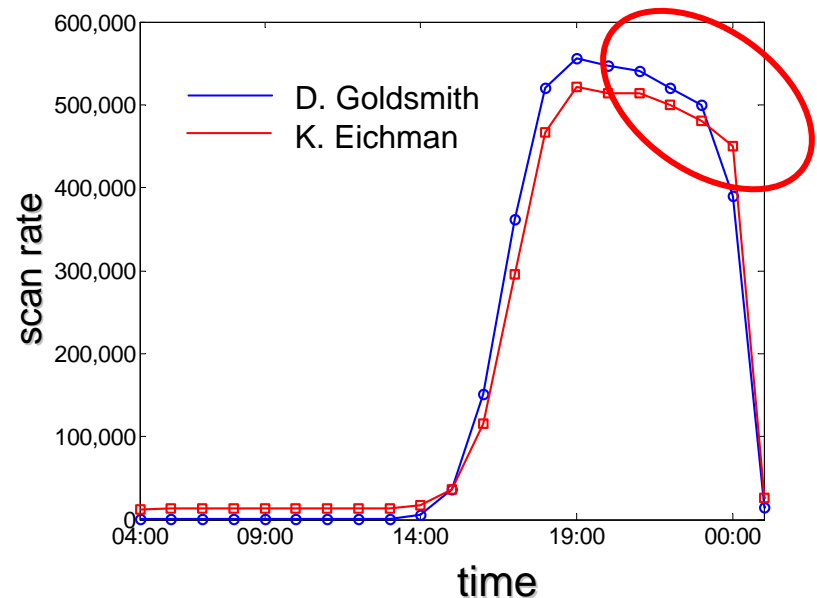
# Code Red: model

- measurements from Class A networks
- simple epidemic model matches increasing part of observed Code Red data. (Staniford)

What about decrease?

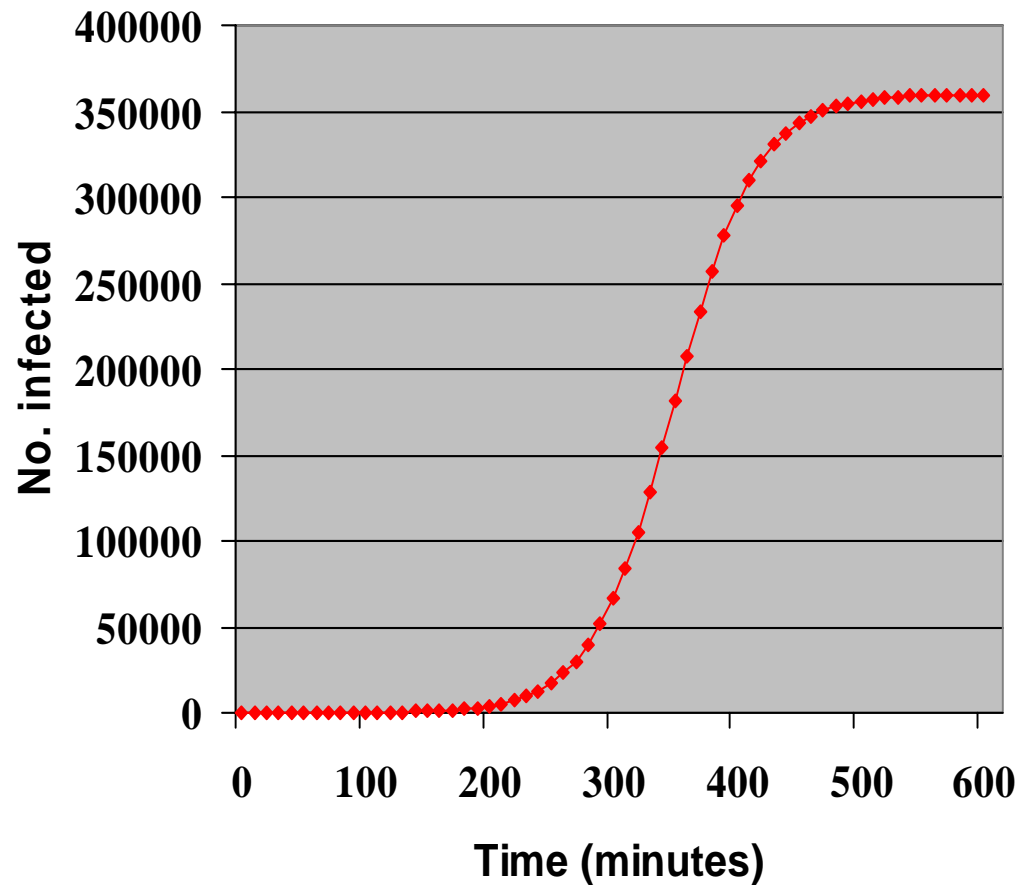
- human countermeasures
- congestion

Zou, etal, 2002



# Assumptions

- classic epidemic model
  - ignore countermeasures
  - ignore congestion
- Code Red parameters
  - $\eta = 358/\text{min}$
  - $N = 360,000$
- uniform scan,  $\Omega = 2^{32}$
- 100s minutes to spread



# Speeding up a worm

- increase  $\eta$
- increase  $I(0)$
- decrease  $\Omega$

# Speeding up a worm

- increase  $\eta$
- increase  $I(0)$
- decrease  $\Omega$

# The perfect worm

- perfect worm

- scan vulnerable nodes exactly once

$$\dot{I}(t) = \beta I(t) \Rightarrow I(t) = \min(e^{\beta t}, N)$$

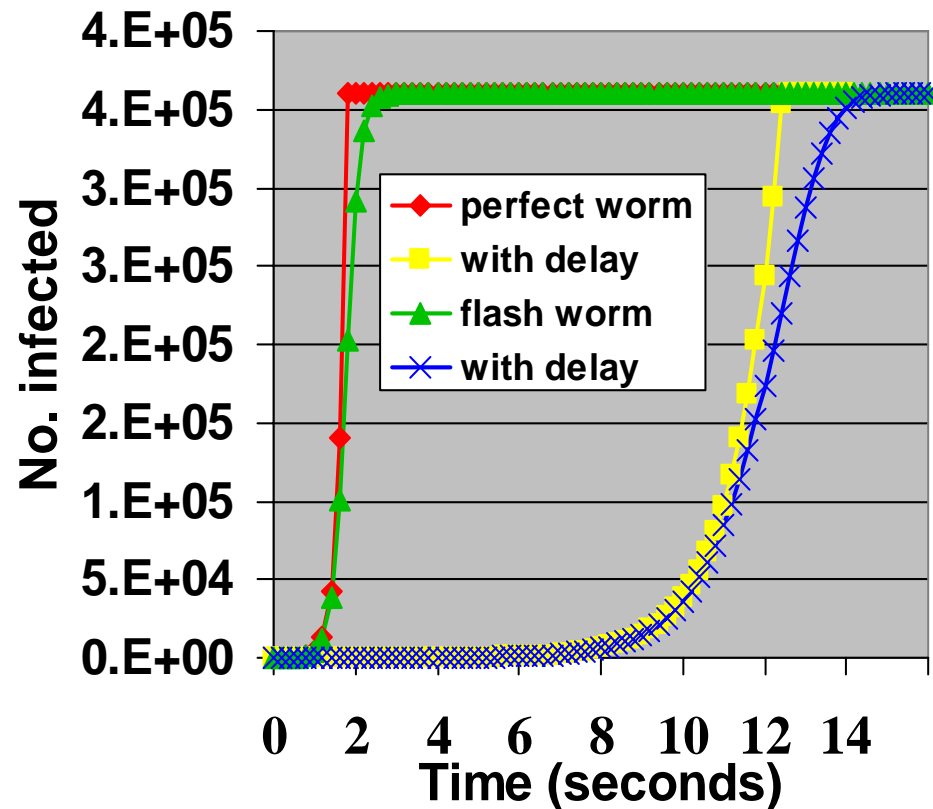
- flash worm (Staniford,...)

- uniform scan of vulnerable nodes ( $\Omega = N$ )

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

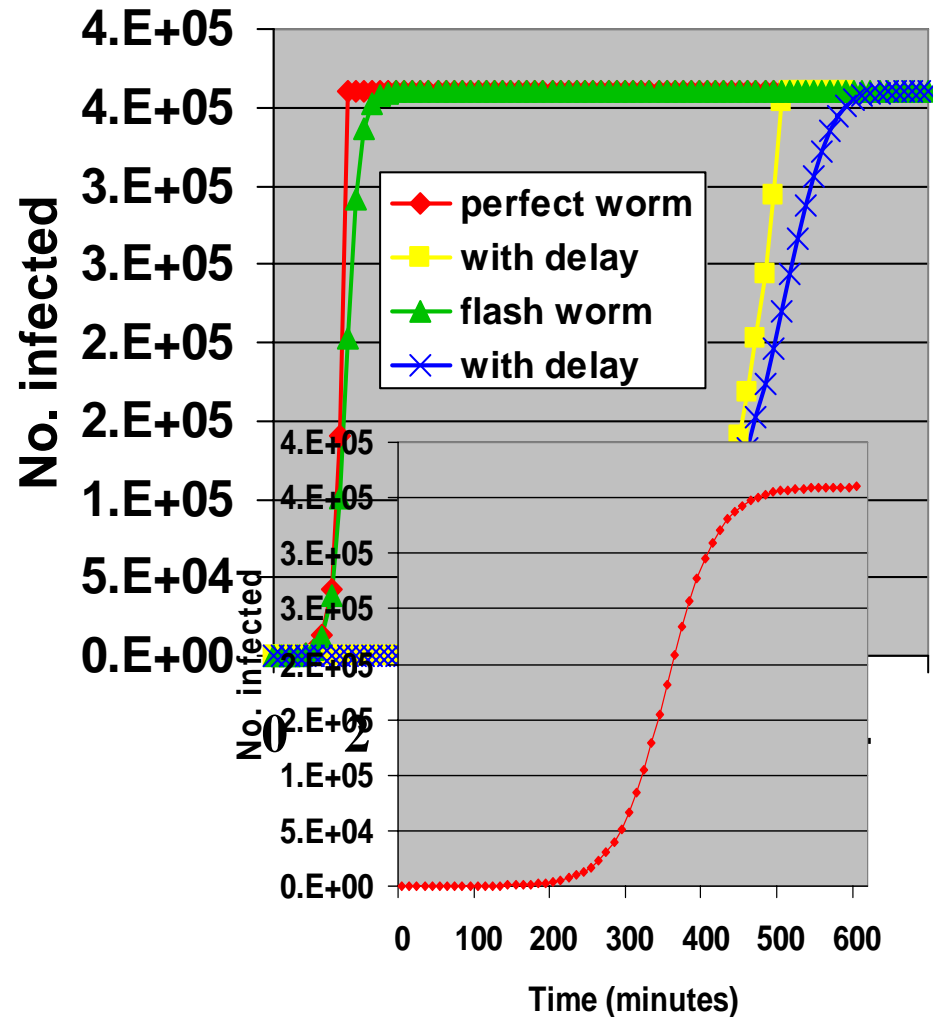
# Perfect Code Red worm

- $I(0) = 10$
- $\eta = 358/\text{min}$
- $N = 360,000$
- all hosts infected within 2 sec.
- 2 second infection delay  
→ six-fold slowdown
- random scan has little effect



# Perfect Code Red worm

- $I(0) = 10$
- $\eta = 358/\text{min}$
- $N = 360,000$
- all hosts infected within 2 sec.
- 2 second infection delay  
→ six-fold slowdown
- random scan has little effect



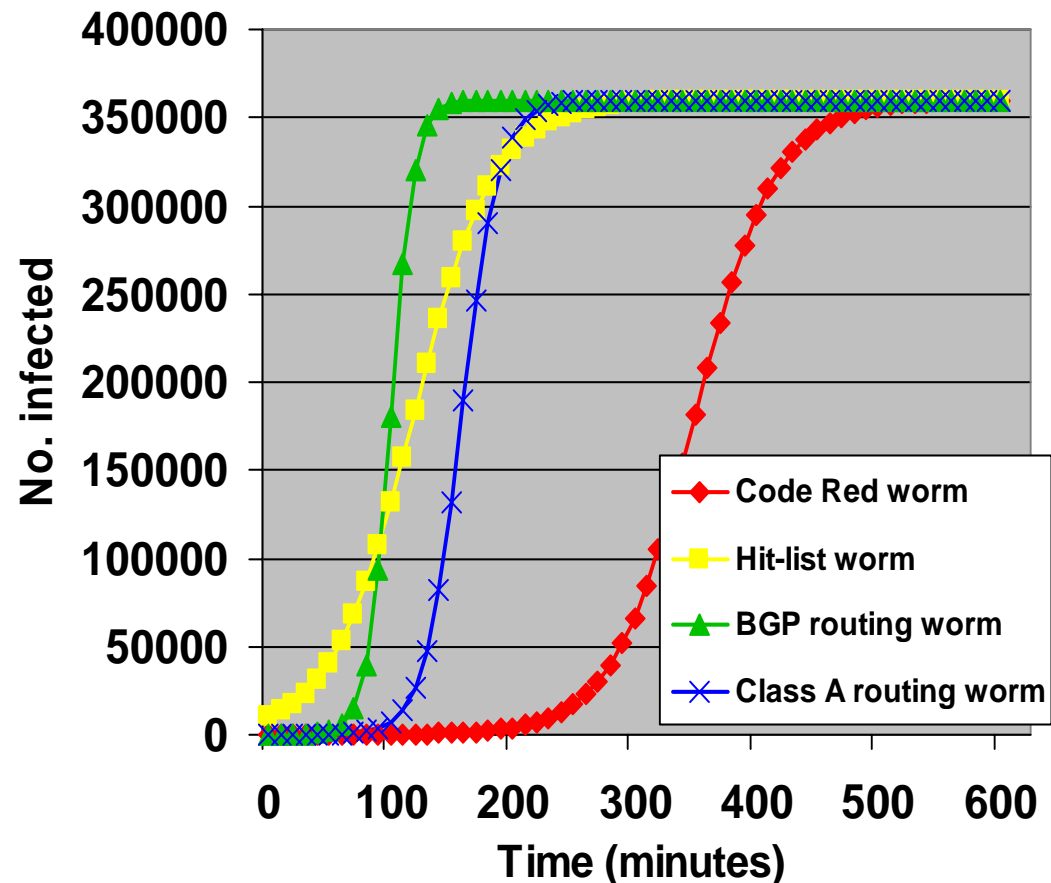
# Hitlist, routing worms

- hitlist worm
  - increase  $I(0)$
- routing worm
  - decrease  $\Omega$
  - BGP table information:  $\Omega = .29 \times 2^{32}$ 
    - 29% of IP address space
  - class A networks:  $\Omega = .45 \times 2^{32}$ 
    - 116 out of 256 possible class A networks

# Hitlist, routing worms

- Code Red style worm
- $\eta = 358/\text{min}$
- $N = 360,000$
- hitlist,  $I(0) = 10,000$
- routing worm as effective as hitlist worm

Payload requirements of routing worm?

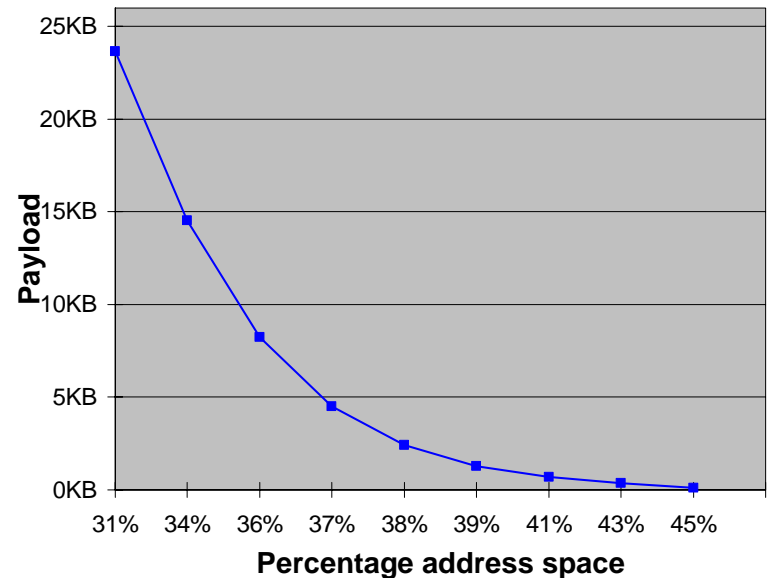
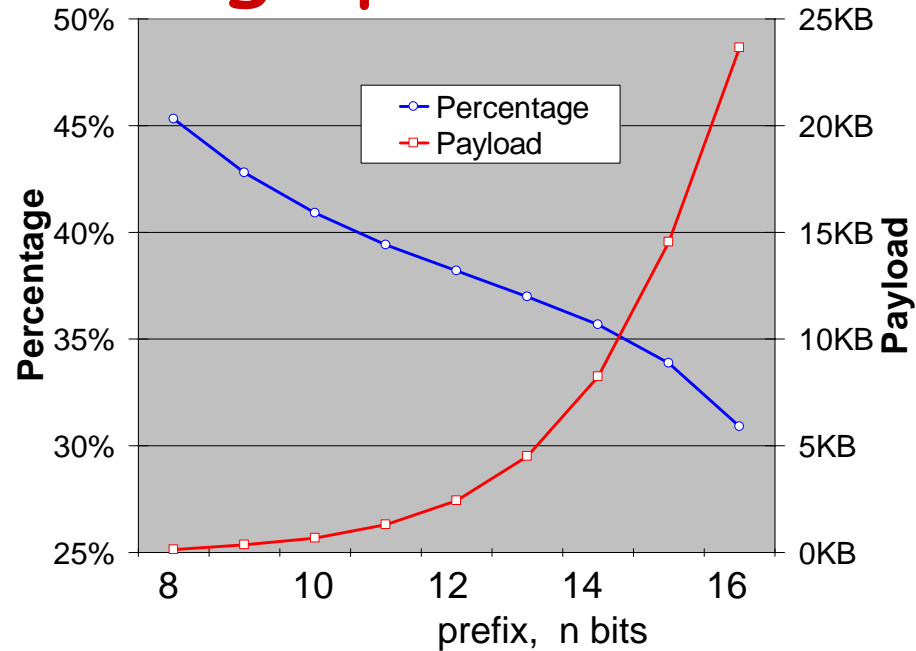


# Payload vs. Scanning space

## Payload

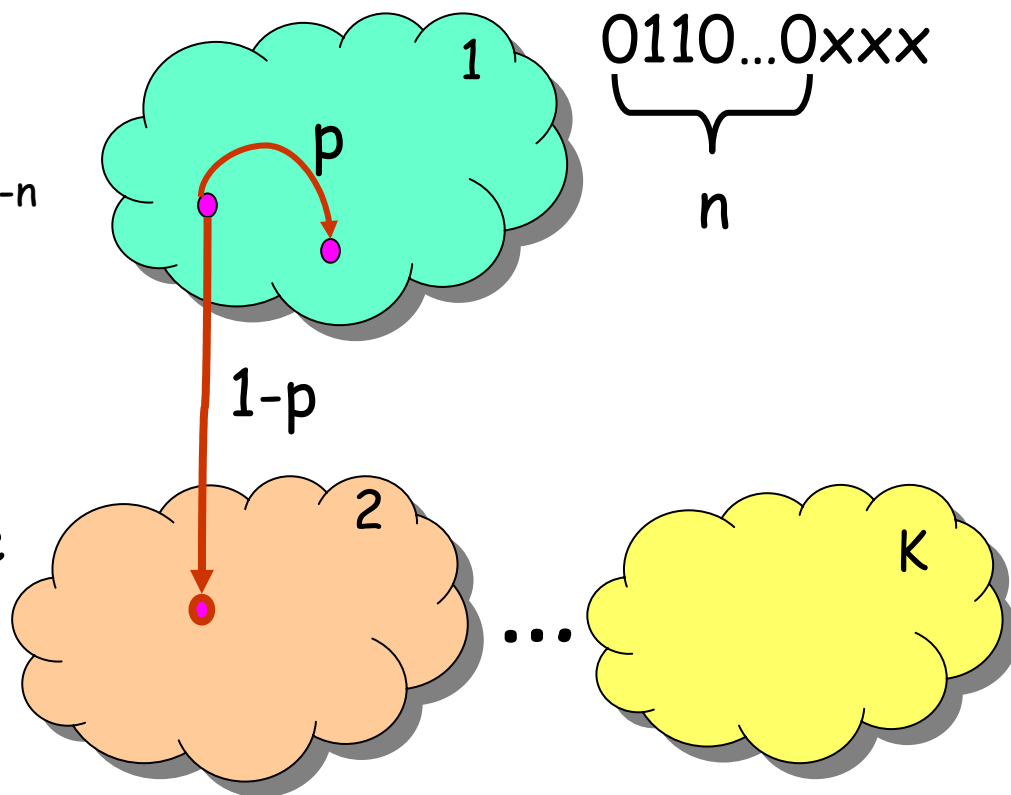
- BGP worm: 62K bytes
- Class A worm: 116 bytes
- “/n” aggregation
  - trade off payload, scanning space

Can one achieve benefit of routing worm w/o payload?



# "/n" Local preference worm

- K "/n" prefix subnetworks,  $\Omega = K \times 2^{32-n}$ 
  - e.g., class A subnets (n=8)
- p - probability scan local prefix subnet
- (1-p) - prob. scan outside local prefix subnet



## /n Local preference worm

- $N_k$ , no. vulnerable hosts in subnet  $k$
- $I_k(t)$ , no. infected hosts in subnet  $k$
- fits epidemic model for interacting groups

## /n Local preference worm

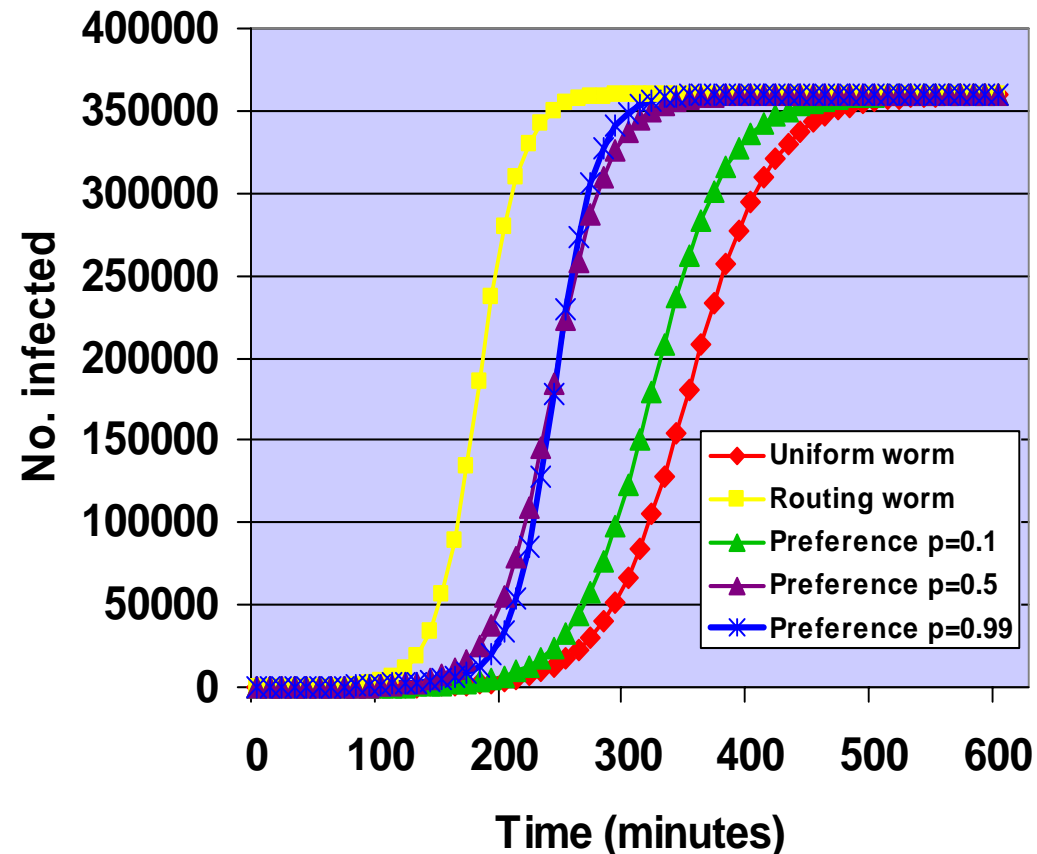
$$\dot{I}_k(t) = \left[ \beta' I_k(t) + \sum_{j \neq k} \beta'' I_j(t) \right] \times [N_k - I_k(t)]$$

- $\beta'$ - local scan rate,  $\beta' = p \eta / 2^{32-n}$
- $\beta''$ - global scan rate,  $\beta'' = (1-p) \eta / ((K-1) 2^{32-n})$
- initial conditions  $I_k(0)$

# /8 Local preference worm

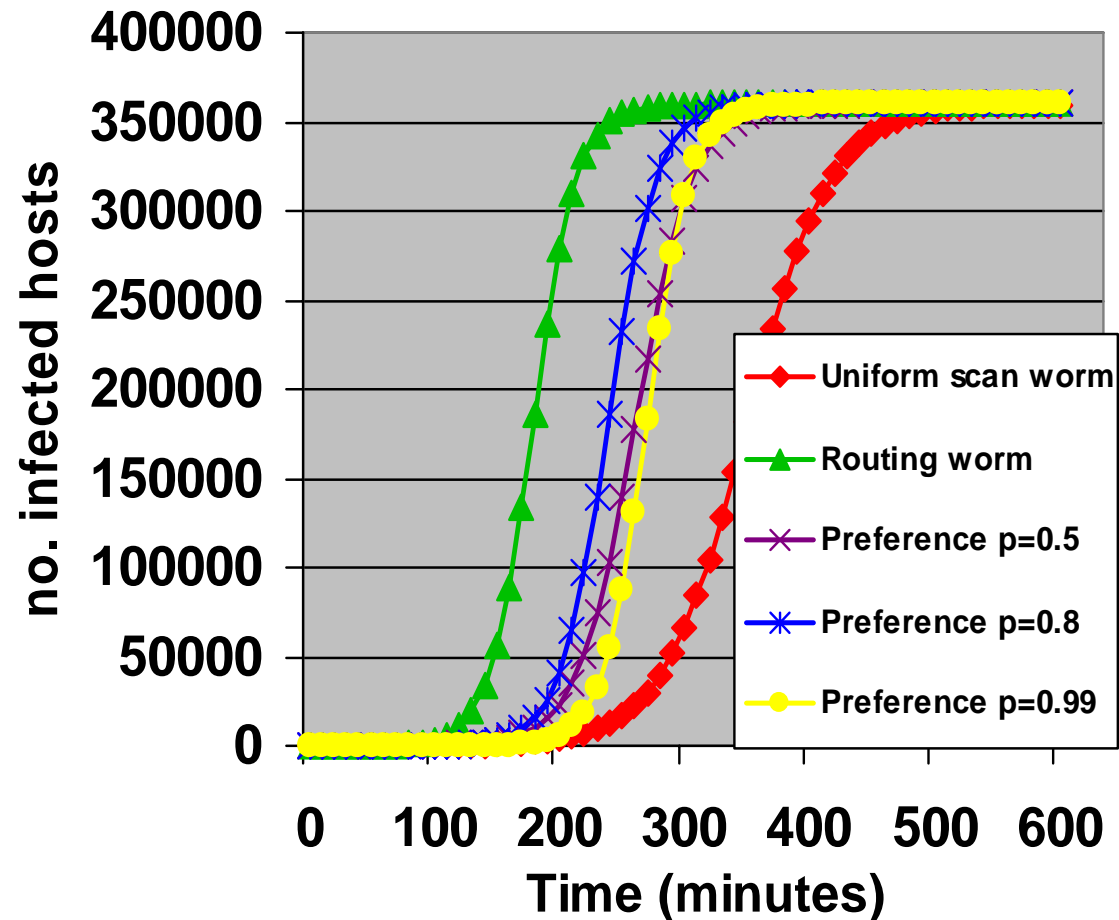
## Class A preference

- $K = 116$
- $N_k = 360,000/K$
- $I_1(0) = 10;$   
 $I_k(0) = 0, k > 1$
- $\eta = 358/\text{min}$
  
- good performance,  
 $p > 0.5$ 
  - $p < 0.5$  for past worms
- approaches speed of routing worm



# /16 Local preference worm

- $K = 29,696$
- $N_k = 360,000/K$
- $I_1(0) = 10;$   
 $I_k(0) = 0, k > 1$
- $\eta = 358/\text{min}$
  
- good performance  
 $p \approx 0.8$
- previous worms have  
 $p < 0.5$



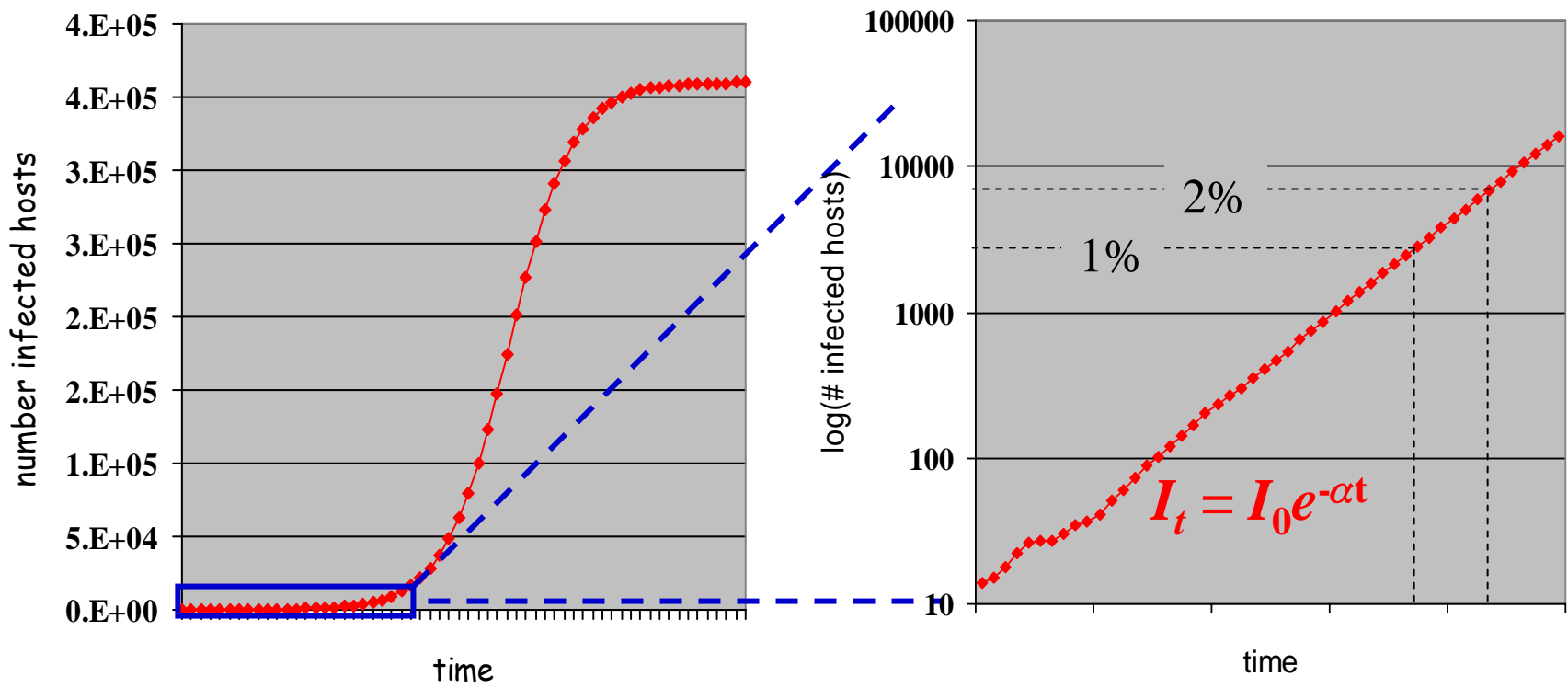
# Sequential vs Random scan

Blaster:

- choose first address randomly
- subsequent addresses chosen sequentially
- average behavior accurately modeled as random scan

# Pause

- deterministic models appropriate for network worms
- worm detection, parameter estimation?
  - $\alpha (\beta N), N$



Initial stage exhibits exponential growth

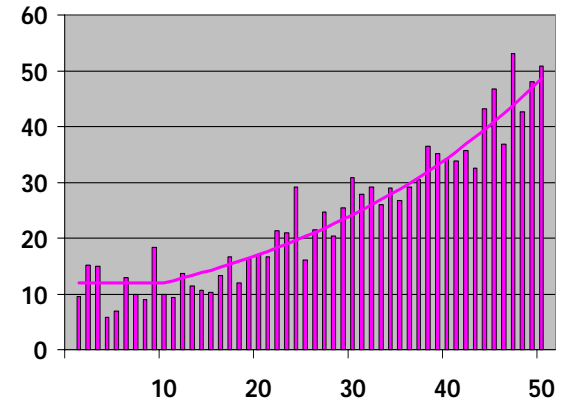
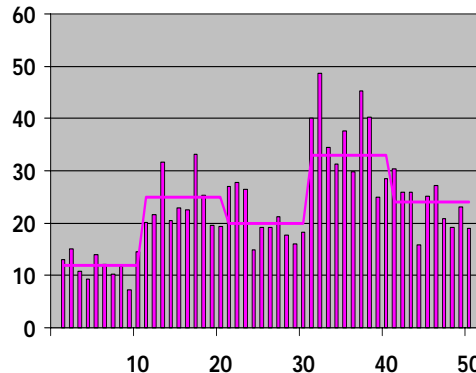
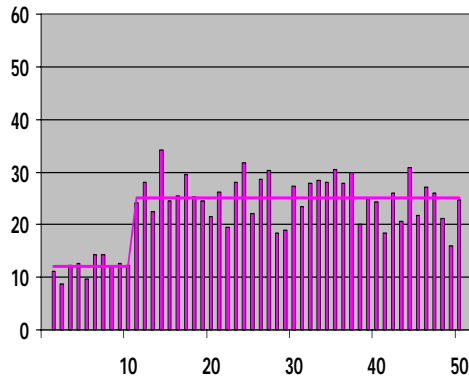
# "Trend Detection"

— Detect traffic *trend*, not *burst*

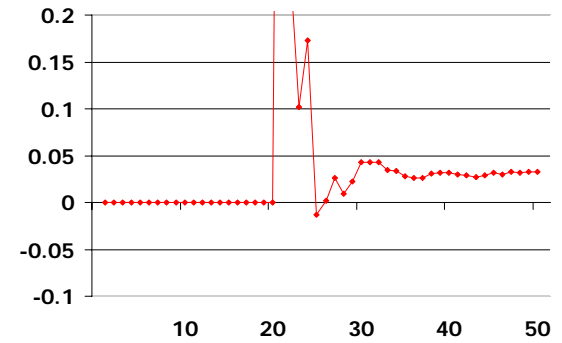
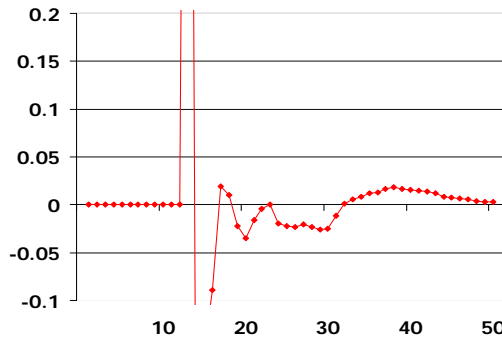
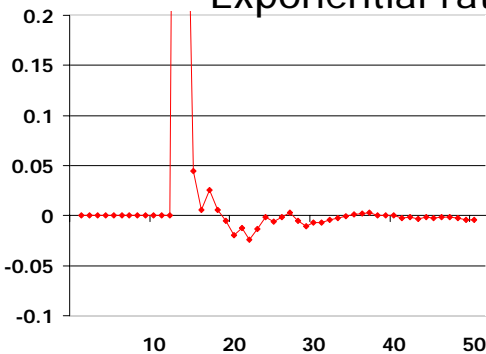
Trend: worm exponential growth trend at the beginning  $\dot{I}(t) = \alpha I(t)$

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

Monitored illegitimate traffic rate



Exponential rate  $\alpha$  on-line estimation



Non-worm traffic burst

Worm traffic

# Approach

- discrete time geometric growth

$$I_t = (1 + \alpha)I_{t-1}, \quad t = 1, 2, \dots$$

or

$$\ln I_t = \ln I_{t-1} + \ln(1 + \alpha), \quad t = 1, 2, \dots$$

- estimate  $\hat{I}_t$  from observed scans
- estimate  $a$  ( $\beta N$ ) from  $\hat{I}_t$
- estimate  $N$  from  $a$

# Worm detection

- collect worm scan traffic
  - distributed monitors
- observations from monitors

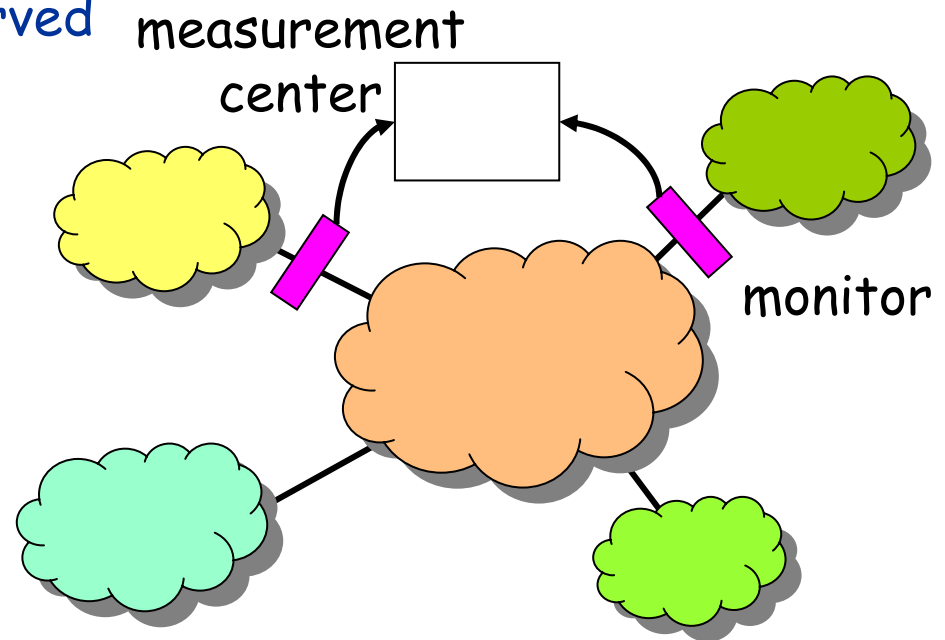
- $C_t$  : cumulative # of observed infected by  $t$ ,

- $Z_t$  : # of scans at time  $t$

- estimate for  $\hat{I}_t$

$$\hat{I}_t = C_t \quad \text{biased}$$

$$\hat{I}_t = Z_t / \eta \quad \text{unbiased}$$



# Estimation of $\alpha$

- stochastic linear recurrence

$$\ln Z_t = \ln Z_{t-1} + \ln(1 + \alpha) + n_t, t = 1, 2, \dots$$

$$\ln Z_t = \alpha t + K + n_t, t = 1, 2, \dots$$

$$n_t \rightarrow 0 \text{ as } t \rightarrow \infty$$

- estimate  $\alpha, K$  with Kalman filter

- estimate  $N, \quad N = \Omega \alpha / \eta$

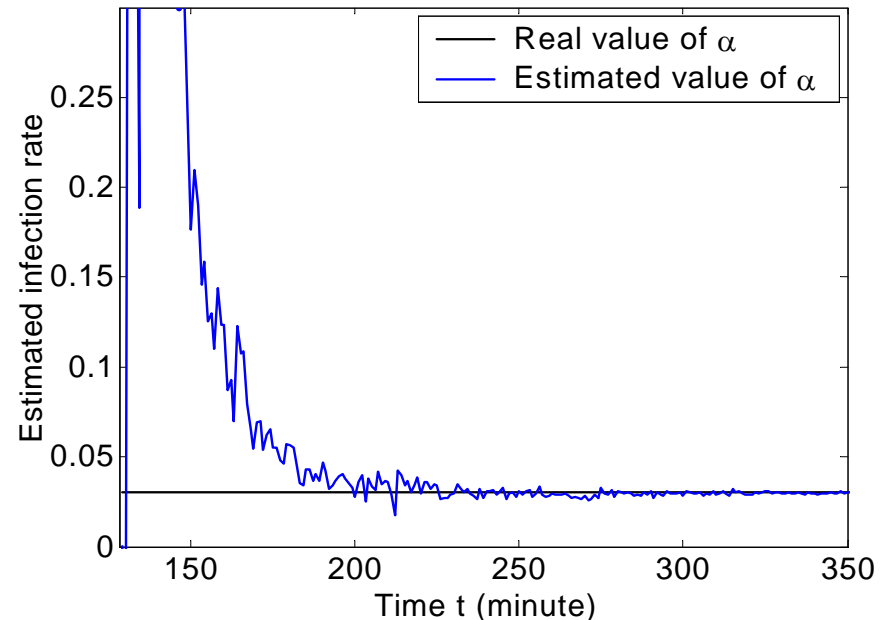
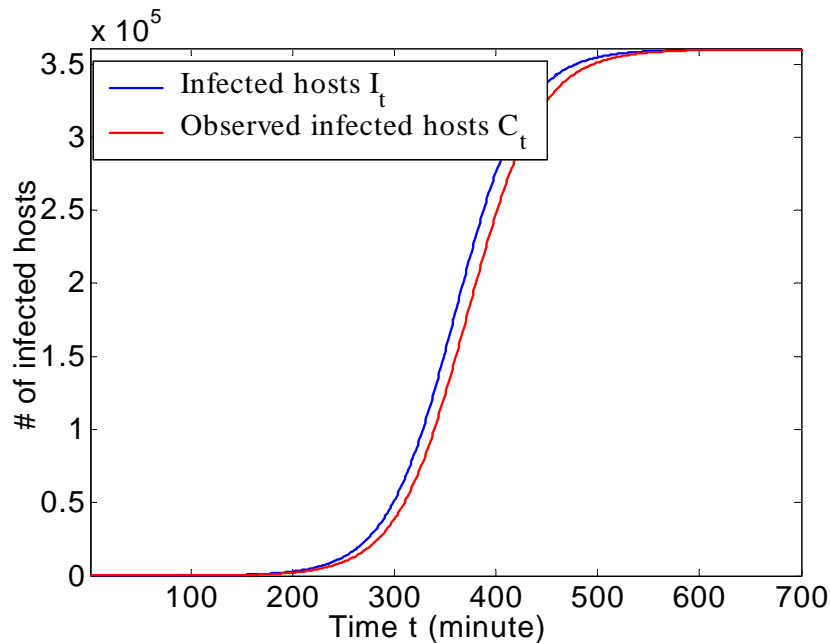
# Worm detection

- monitor scan traffic  $Z_t$
- if  $Z_t > \text{threshold}$   
then start Kalman filter
- if estimate of  $\alpha$  stabilizes at value  $> 0$   
then worm present

# Code Red simulation experiments

Population:  $N=360,000$ ,  
Scan rate  $\eta = N(358/\text{min}, 100^2)$ ,  
Monitored IP space  $2^{20}$ ,  
Background scans

Infection rate:  $\alpha = 1.8/\text{hour}$ ,  
Initially infected:  $I(0) = 10$   
Monitoring interval: 1 minute

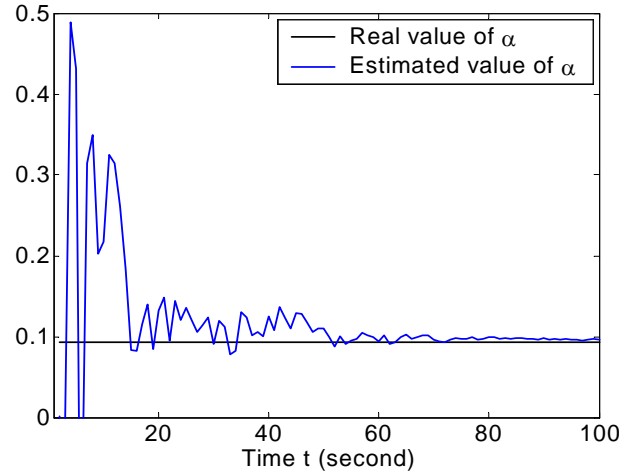
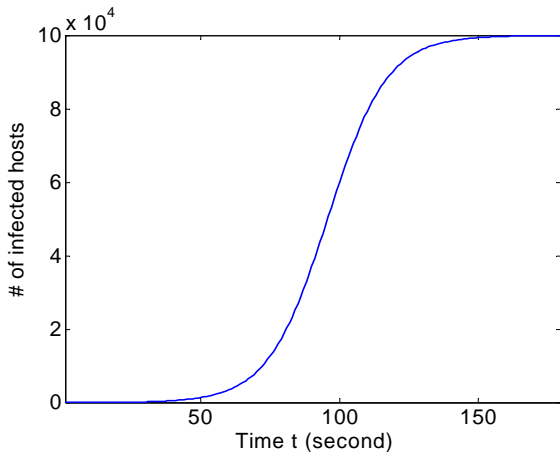


At 1% (220 min): **estimate stabilizes, oscillates around  $\alpha$**

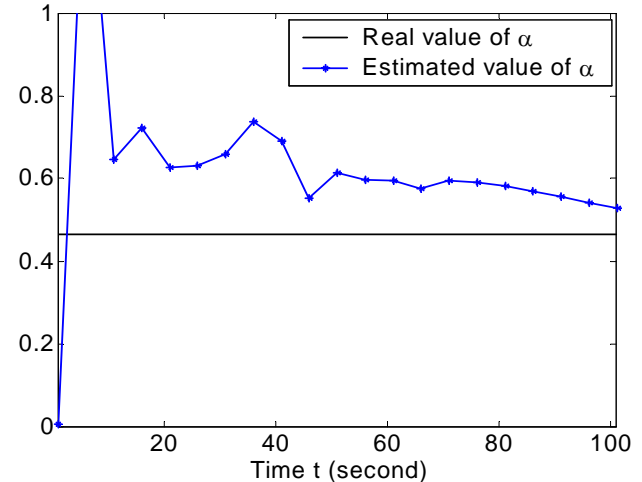
# SQL Slammer Experiments

Population:  $N=100,000$ ,  
Initially infected  $I_0=10$

Scan rate  $\eta = N(4000/\text{sec}, 2000^2)$ ,  
Monitored IP space  $2^{20}$



Monitoring interval: 1 second



Monitoring interval: 5 second

□ at 1% (47 seconds): relatively stabilized estimation

□ larger monitoring interval:

- difficult to see trend; easier to implement
- increased discrete-time model error
- use delayed estimation with small interval

# Issues

□ estimator for  $\alpha$  assumes random scans

Q: how do we make sequential scans look random?

A: divide monitored space into small chunks randomly spread around address space

# Current Efforts

- more efficient estimation
- distributed estimation
- automatic quarantine in enterprise networks

# Summary

- simple models useful for worm studies
- routing worms most virulent
  - local preference worms approach routing worms
- look for exponential trends
- detection/estimation possible in early stages of worm spread

Challenge: what to do with knowledge?

# References

- C.C. Zou, D. Towsley, and W. Gong. "On the Performance of Internet Worm Scanning Strategies", *Umass ECE Technical Report TR-03-CSE-07*, November, 2003.  
<http://tennis.ecs.umass.edu/~czou/research/wormStrategy-techreport.pdf>
- C.C. Zou, L. Gao, W. Gong, D. Towsley. "Monitoring and Early Warning for Internet Worms". *10th ACM Conference on Computer and Communication Security (CCS'03)*, Oct. 27-31, Washington DC, USA, 2003.  
<http://tennis.ecs.umass.edu/~czou/research/monitoringEarlyWarning.pdf>