Botnets for scalable management

Jérôme François, Radu State and Olivier Festor

MADYNES Loria Nancy-Université

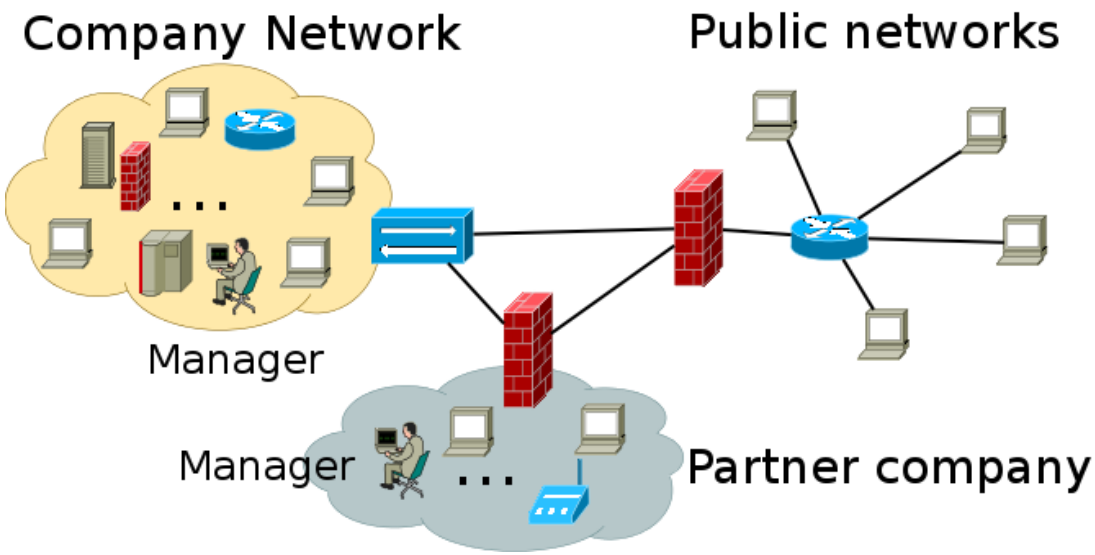
Outline

Outline

Network management challenges

- Increased number of devices to be managed
- Increased diversity of devices to be managed
- Not well delimited domains
- Multiple obstacles: addresses translators, firewall...

$\longrightarrow$ reach many hosts wherever there are



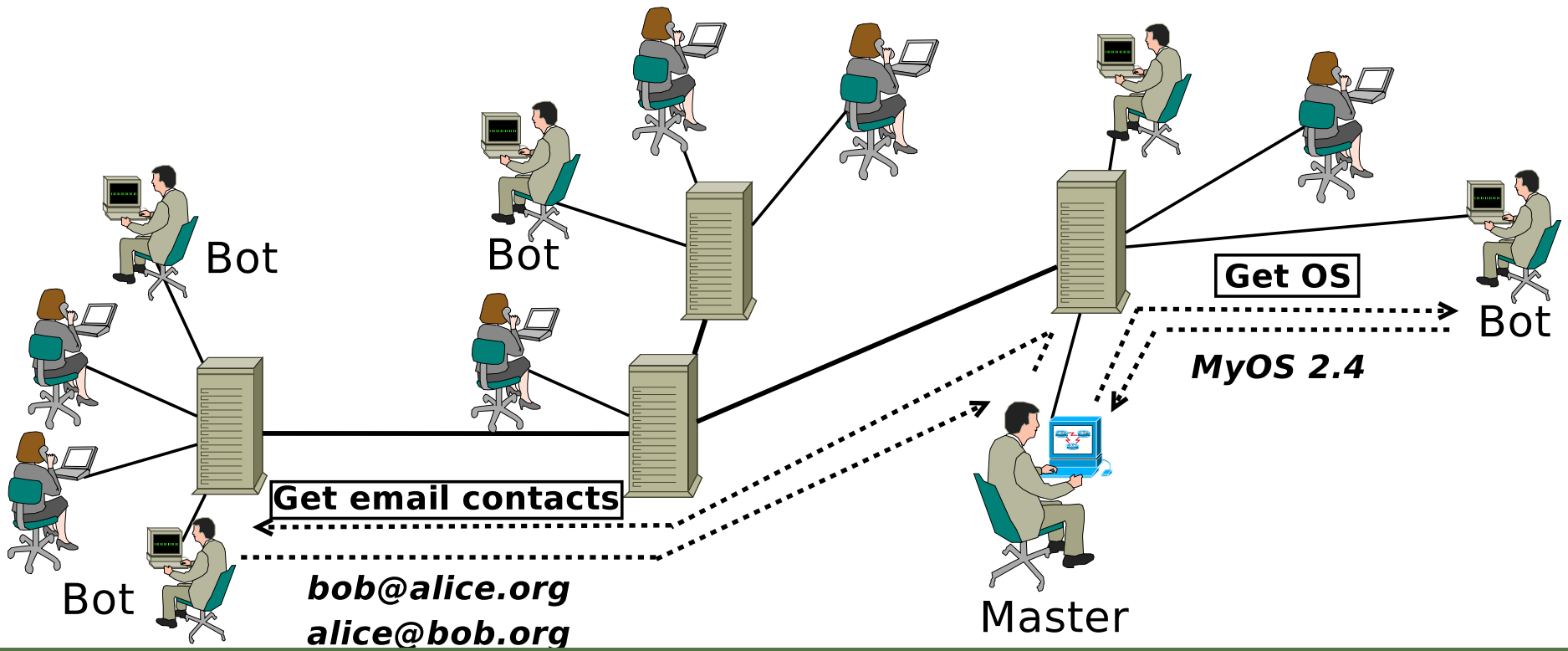
Outline

Malware communication paradigms

- attackers faced the same problems
  - control multiple machines through the Internet
  - goals: distributed denial of service attacks, mass collecting of sensitive data
- worm capabilities
  - infection of multiple machines by various ways
  - execution of a payload
- construction of a botnet
  - control mechanism to send orders to the bots and get the responses
  - decentralized and scalable: example of 400 000 zombies in one botnet
  - bypass most of the security equipements due to the self connection of zombies (outgoing connection)

Botnet based network management

- use a botnet to perform management operations
- IRC (Internet Relay Chat) protocol: chatting
- decentralized architecture, spanning tree



Botnet performances

- current knowledges: observation of malicious activities
- scalability proof needed
- main goal: send a management request to multiple devices with the minimal delay

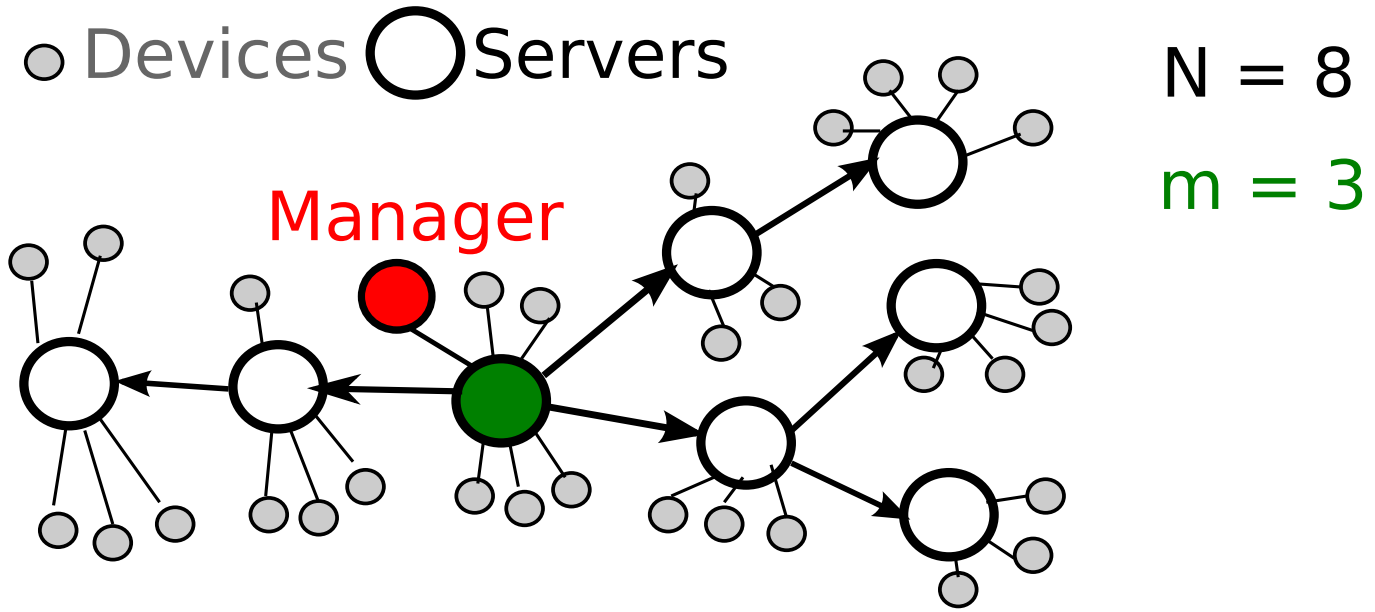
Addressed questions:

- what is the probability to reach 80% of the devices ?
- how much time is needed to perform the operations ?
- how many servers should be deployed ?
- what is the best topology ?

Outline

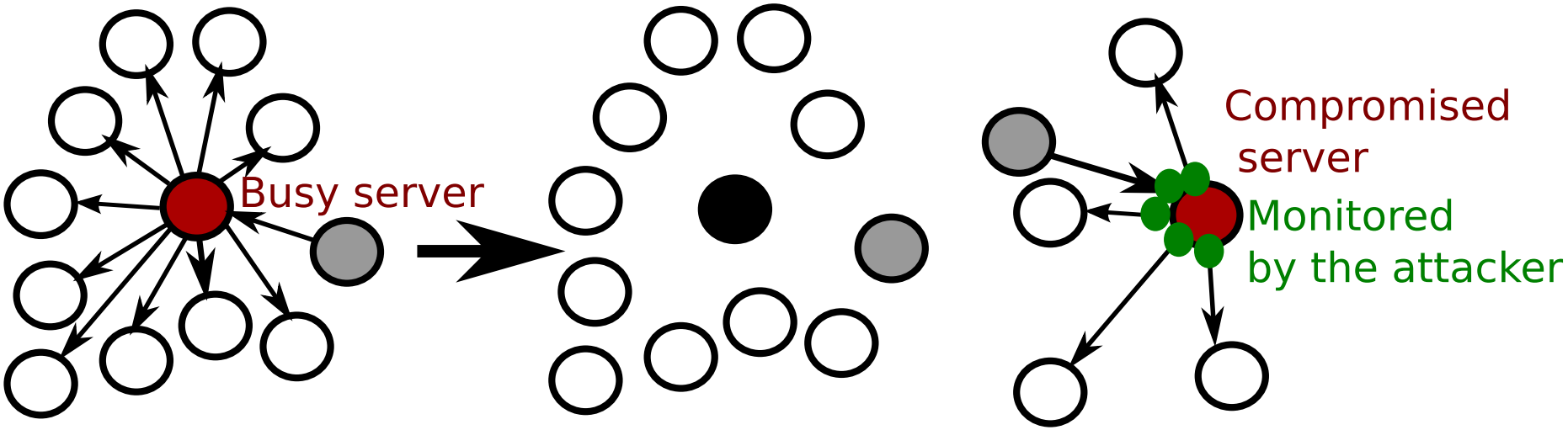
Parameters

- consider only the servers
- N: total number of servers
- m is the maximal branching factor which is equal to the maximal number of neighbors

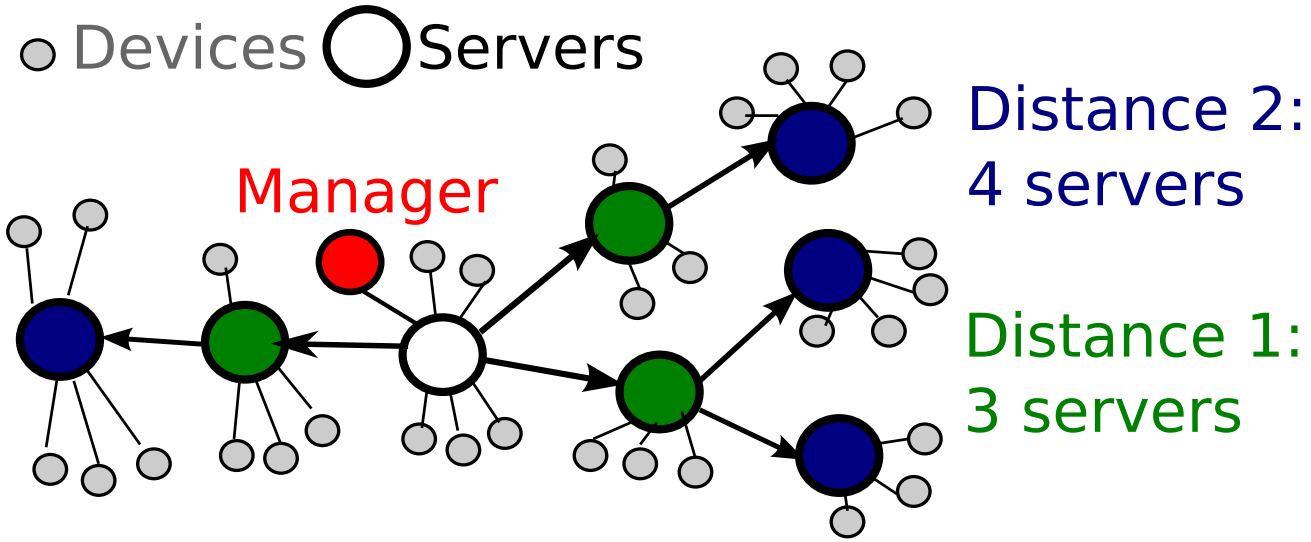


Parameters

- a server can crash if it has to maintain too many connections $\longrightarrow \alpha(m)$ is the probability for a server to be able to forward the messages, decreasing function
- the risk to be discovered by an attacker and to be attacked: β



Goal: determine the reachability = the number of servers reached at a certain distance



- a server can have between 0 and m neighbors
- the probability p_k for a server to have k neighbors

$$p_k = \begin{cases} 1 - \alpha(m) & \text{when} \quad k = 0 \ (\textit{overloading}) \\ \dfrac{\alpha(m)}{m} & \text{when} \quad 1 \leq k \leq m \ (\textit{random}, \textit{uniform}) \\ 0 & \text{when} \quad k > m \end{cases}$$

The generating functions

For p_k: $G_0(x) = \sum_{k=0}^{\infty} p_k x^k$

- $k = 0 \rightarrow p_0$
- $k = 1..m \rightarrow p_k x^k = \frac{\alpha(m)}{m} x^k$
- $k = m + 1...\infty \rightarrow 0$
- $G_0(x) = p_0 + \frac{\alpha(m)}{m} \times \sum_{k=1}^{m} x^k$

Useful function to compute the average number of neighbors: $\mathbb{E}(k) = \frac{dG_0}{dx}(1)$

The generating functions

- average number of servers at the *jth* hop[1]

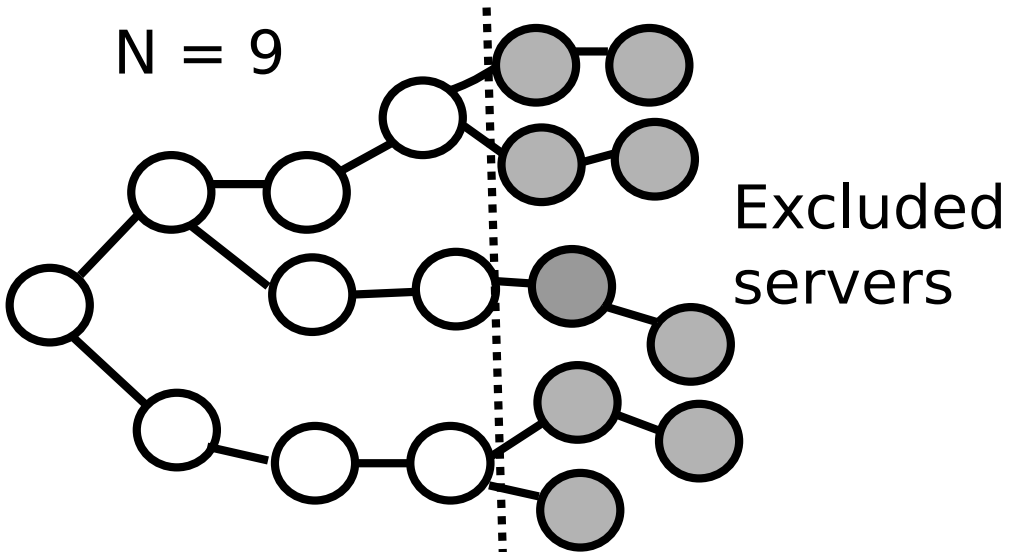
$$z_1 = \frac{dG_0}{dx}(1) \quad z_2 = \frac{d^2G_0}{d^2x}(1) \quad z_j = \left[\frac{z_2}{z_1}\right]^{j-1} z_1$$

[1] Ramachandran K. and Sikdar B.: Modeling malware propagation in gnutella type peer-to-peer networks. Parallel and Distributed Processing Symposium, 2006

Reachability

- number of reached servers at a maximal distance k divided by the total number of servers

$$reachability(k) = \frac{min(\sum_{j=1}^{k} z_j, N)}{N}$$



Reachability

- one compromised server $\longrightarrow$ all the network is compromised
- $(1-\beta)^N =$ probability to have no compromised servers
- the number of reached servers is limited by the probability to have a functionning network

$$reachability(k) = \frac{(1-\beta)^N \times min(\sum_{j=1}^{k} z_j, N)}{N}$$

Average reachability

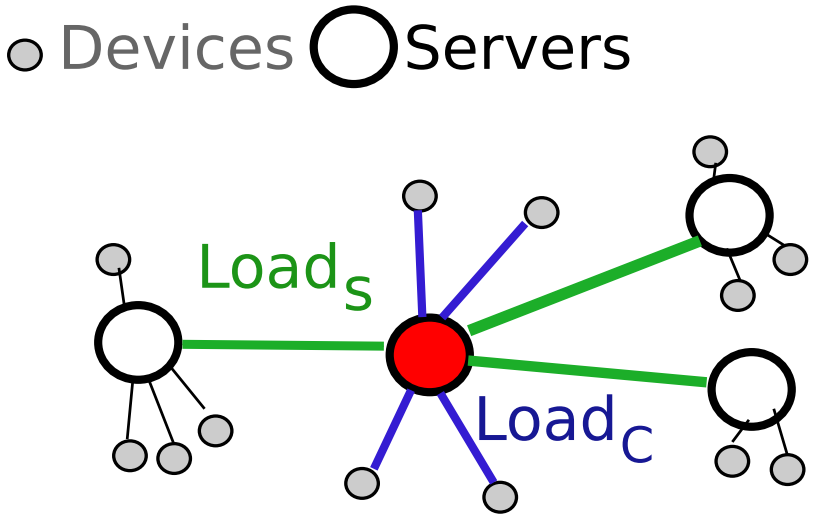
- average reachability over all possible distance in a tree

$$avg\_reachability(k) = \frac{\sum_{k=1}^{N} reachability(k)}{N}$$

- random uniform connection of bots $\longrightarrow$ same values for bots

The load of the system

- $load_s$: load of a server to maintain connections with other servers
- $load_c$: load of a server to deal with the connected hosts



The load of the system

- C devices in the system
- C_{server} is the average number of bots per server.
- simple and centralized approach:

$$load\_server_{centralized} = C \times load_c$$

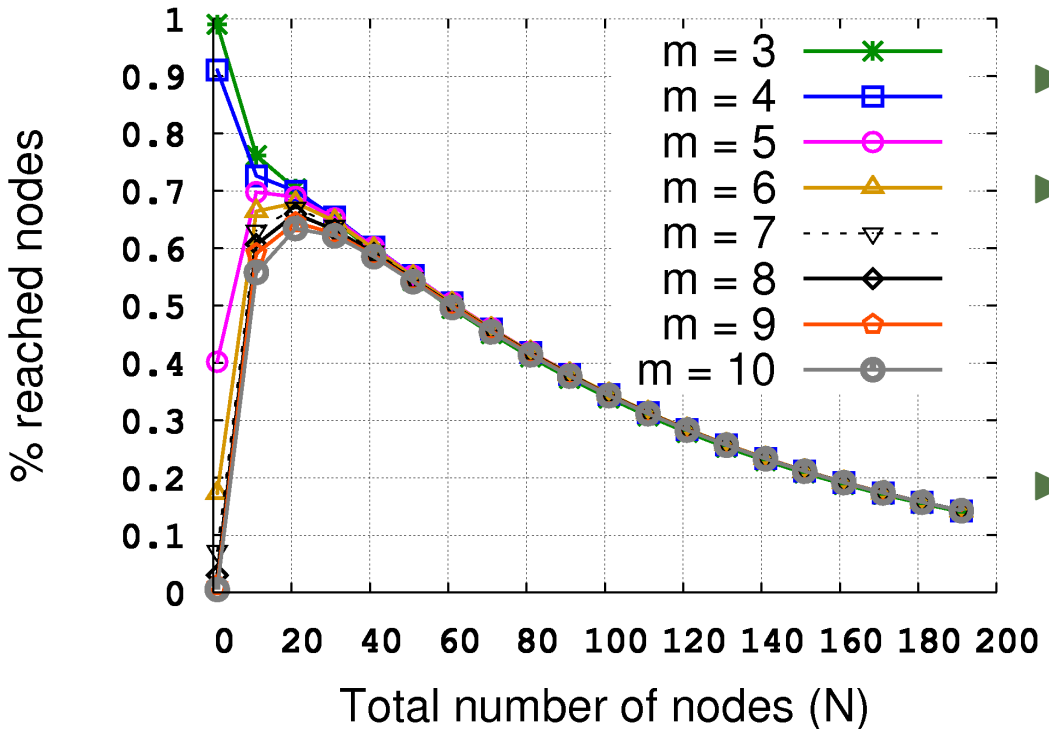
- our solution:

$$load\_server_{botnet} = C_{server} \times load_c + m \times load_s$$

Outline

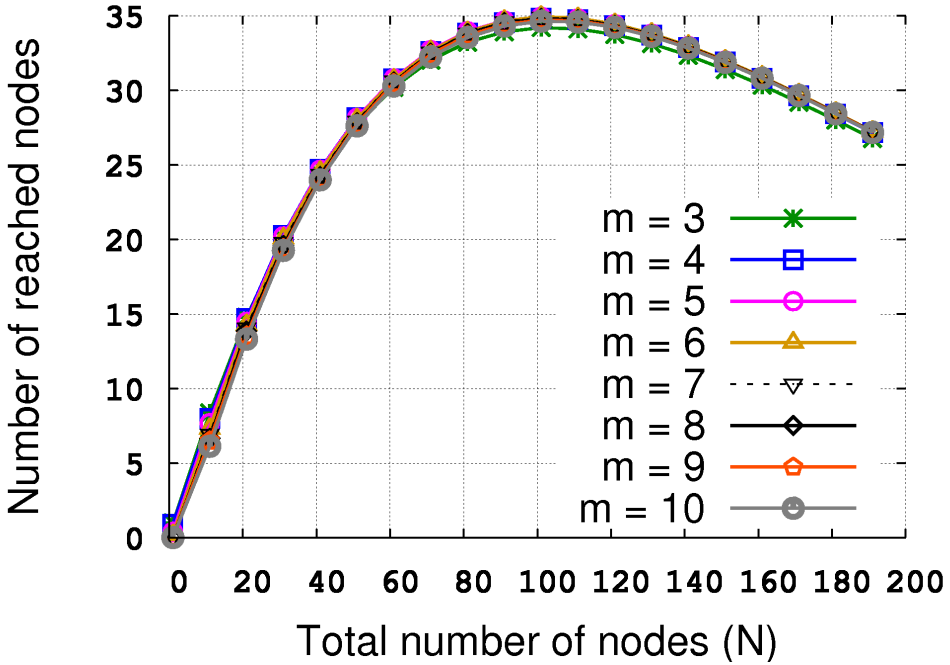
Average Reachability

- $\beta = 0.01$
- $\alpha(m) = e^{(3-m)}$



- N varies from 1 to 200
- m between 3 and 10
- tree = chain for $m = 2$
- small branching factor $\longrightarrow$ better reachability $\longrightarrow$ $\alpha(m)$ impact
- using more than 20 servers is useless for reachability

Absolute reachability

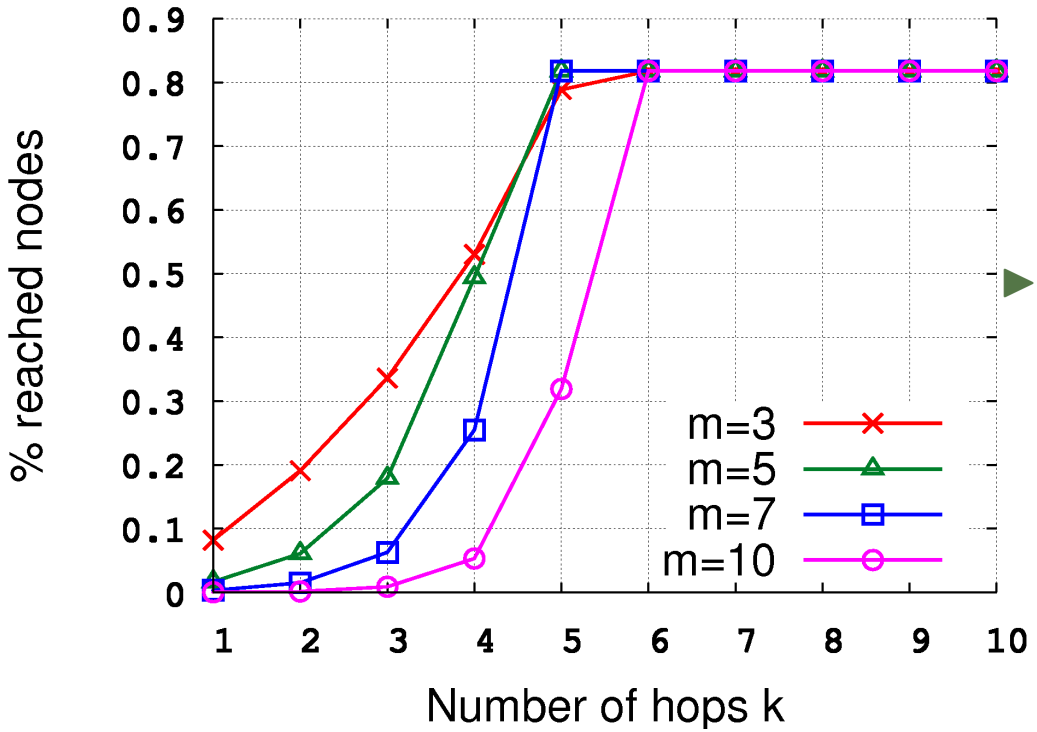


- goal: reach the maximum number of servers
- more than 100 servers is totally ineficient
- 35 servers can be reached in the best case

- 100 hosts per server $\longrightarrow$ 3500 hosts
- $load_{centralized} = 3500 \times load_{host}$
- $load_{botnet} = 100 \times load_{host} + m \times load_{server}$
- botnet is more efficient if $load_{server} < \frac{3400}{m} load_{host}$

Reachability and the number of hops

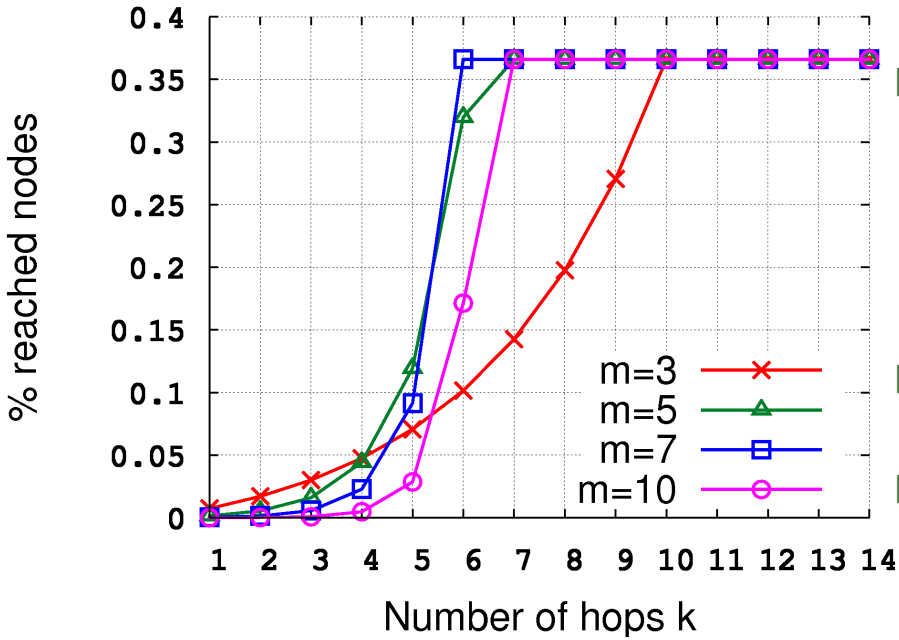
- The number of hops varies from 1 to 10



- N = 20 servers
- $m \in 3, 5, 7, 10$
- impact of $\alpha(m)$: high connectivity and low $N \rightarrow$ low reachability

Reachability and the number of hops

- The number of hops varies from 1 to 10
- N = 100 servers
- $m \in 3, 5, 7, 10$

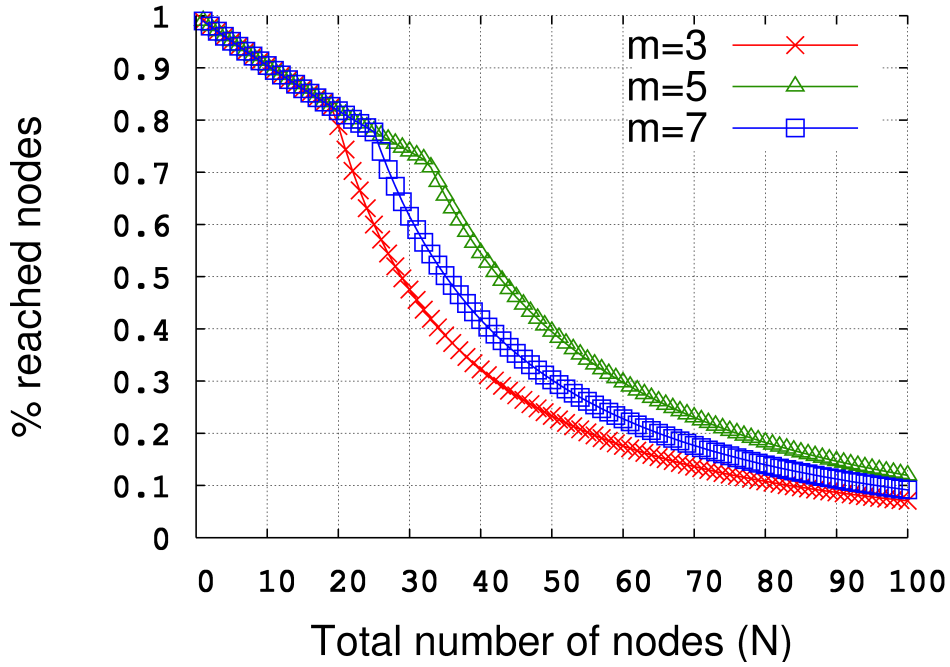


- lowest reachability for $m = 3$ $\longrightarrow$ m has a greater impact than $\alpha(m)$ due to the value of N
- m and $\alpha(m)$ are antagonist
- important curves to determine the servers topology

- $m = 5$ is a good choice for a 20-100 servers network
- maximum reachability $\longrightarrow$ 7 hops

Reachability and the number of nodes

- distance = 5 hops



- N varies from 1 to 100
- first stage limited by β, second stage limited by $\alpha(m)$
- competitive impact of $\alpha(m)$ and $\beta \longrightarrow$ one limitative factor at a certain time

Outline

Current solutions

- centralized solution
- decentralized solutions with mid-level managers (requests translations, results aggregation)
- management by delegation
- active networks
- salutary worms: CodeRed / CodeGreen

Advantages of IRC botnets for network management

- the servers have just to forward the messages (light software)
- the IRC clients for the devices do not need many resources
- the devices have to connect themselves to the botnet $\longrightarrow$ bypass active network equipments

Topology and Scalability

Evaluation $\longrightarrow$ help an administrator to choose the right topology

- maximum number of servers: 100, 20 for the best "profitability of servers"
- 100 servers $\longrightarrow$ 35 reached servers (35%)
- 20 servers $\longrightarrow$ 16 reached servers (81%)
- maximal distance: 7 hops
- best branching factor: $m = 5$
- 200 bots per servers $\longrightarrow$ the same management operation can be sent to 7000 devices simultaneously

Future works

- implementation and deployment of the solution on a large testbed (PlanetLab, EmanicsLab)
- case study: configure a large distributed honeypot for detecting cyber predators on P2P file sharing systems
- study the other botnets communication mechanisms

Botnets for scalable management

Jérôme François, Radu State and Olivier Festor

MADYNES Loria Nancy-Université