Detecting Service Violation in Internet and Mobile Ad Hoc Networks

Bharat Bhargava
CERIAS Security Center
CWSA Wireless Center
Department of CS and ECE
Purdue University
bb@cs.purdue.edu

Supported by NSF IIS 0209059, NSF IIS 0242840 ,

NSF CNS 0219110, CISCO, Motorola, IBM

Research Team

- Faculty Collaborators
  - Dongyan Xu, Middleware and privacy
  - Mike Zoltowski, Smart antennas, wireless security
  - Sonia Fahmy, Internet security
- Postdoc
  - Lezsek Lilien, Privacy and vulnerability
  - Xiaoxin Wu, Wireless security
  - Jun Wen, QoS
  - Mamata Jenamani, Privacy
- Ph.D. students
  - Ahsan Habib, Internet Security
  - Mohamed Hefeeda, Peer-to-Peer networking
  - Yi Lu, Wireless security and congestion control
  - Yuhui Zhong, Trust management and fraud
  - Weichao Wang, Security in wireless networks

More information at http://www.cs.purdue.edu/people/bb

Motivation

- Lack of trust, privacy, security, and reliability impedes information sharing among distributed entities.
- Research is required for the creation of knowledge and learning in secure networking, systems, and applications.

Goal

- Enable the deployment of secure applications in the pervasive computing and communication environments.

Objective

- A trustworthy, secure, and privacy preserving network platform must be established for trusted collaboration. The fundamental research problems include:
  - Trust management
  - Privacy preserved collaborations
  - Dealing with a variety of attacks in networks
  - Intruder identification in ad hoc networks
  - Trust-based privacy preservation for peer-to-peer data sharing

Applications

- Guidelines for the design and deployment of security sensitive applications in the next generation networks
  - Data sharing for medical research and treatment
  - Collaboration among government agencies for homeland security
  - Transportation system (security check during travel, hazardous material disposal)
  - Collaboration among government officials, law enforcement and security personnel, and health care facilities during bio-terrorism and other emergencies

A. Trust Formalization

- Problem
  - Dynamically establish and update trust among entities in an open environment.
- Trust based on
  - Evidence
  - Credential
  - Interactions
  - Fraud potential
  - Privacy requirement
- Measure of trust

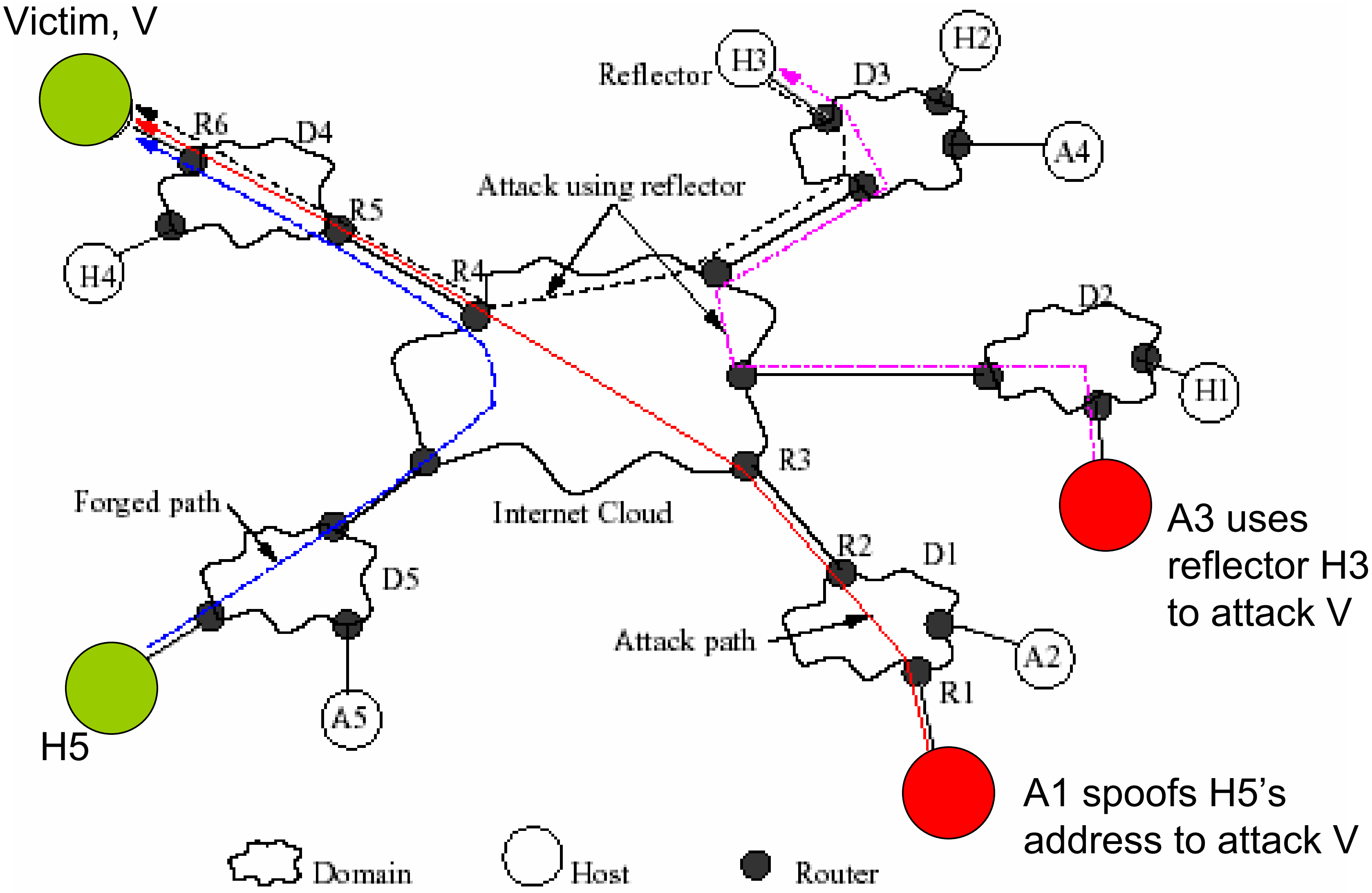
B. Privacy Preserved Collaborations

- Problem
  - Preserve privacy, gain trust, and control dissemination of data
- Privacy based on
  - Approximate location
  - Approximate version of information
  - Any cast
- Determine the degree of data privacy
  - Size of anonymity set metrics
  - Entropy-based metrics
- Tradeoff between privacy and trust

C. Detecting Service Violation in Internet

- Problem statement
  Detecting service violation in networks is the procedure of identifying the misbehaviors of users or operations that do not adhere to network protocols.

Topology Used (Internet)



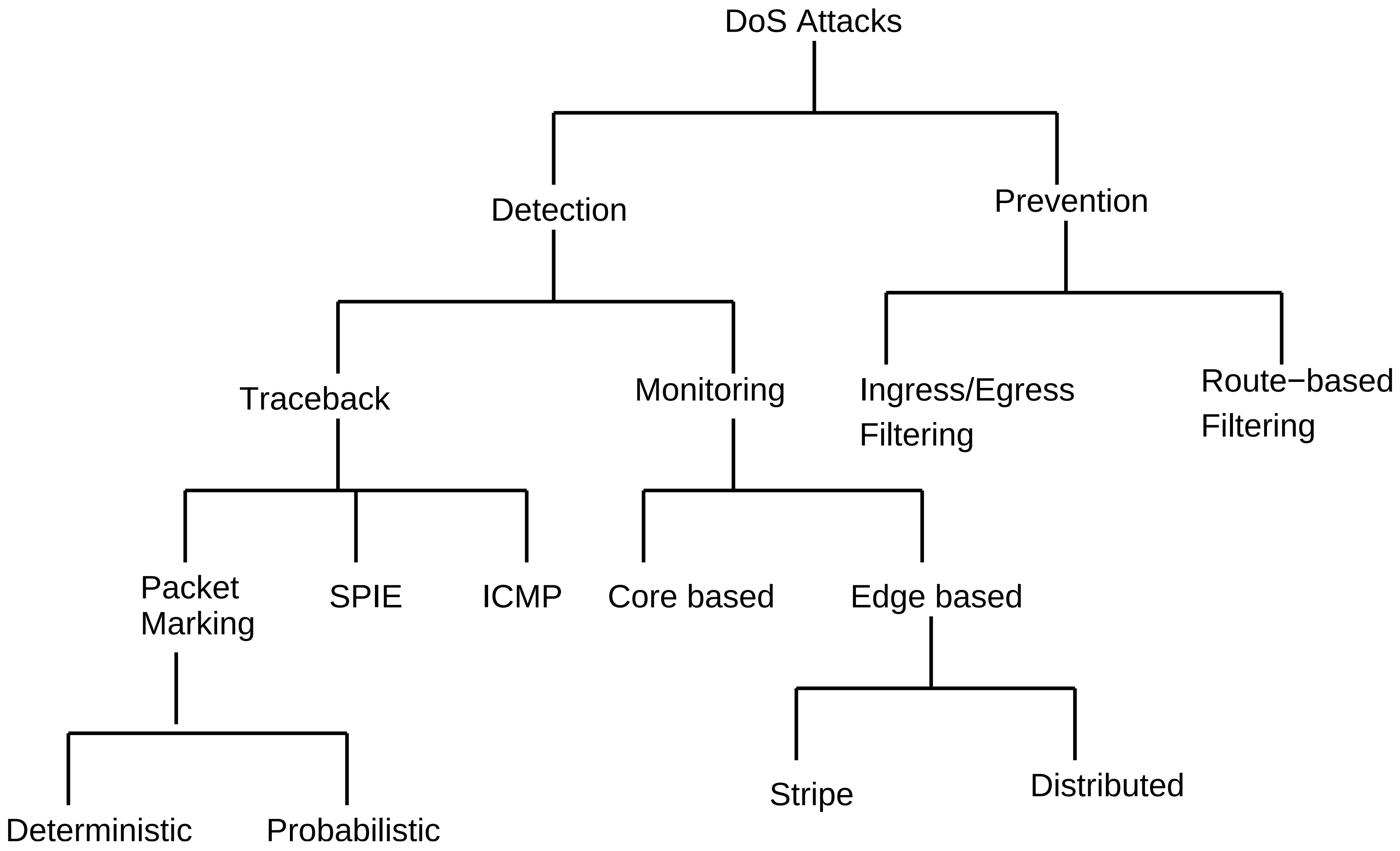
Victim, V

H5

A3 uses reflector H3 to attack V

A1 spoofs H5’s address to attack V

Detecting DoS Attacks in Internet



DoS Attacks

Detection

Prevention

Traceback

Monitoring

Ingress/Egress Filtering

Route−based Filtering

Packet Marking

SPIE

ICMP

Core based

Edge based

Deterministic

Probabilistic

Stripe

Distributed

*SPIE: Source Path Isolation Engine

- Research Directions
  - Observe misbehavior flows through service level agreement (SLA) violation detection
  - Core-based loss
  - Stripe based probing
  - Overlay based monitoring

Approach

- Develop *low overhead* and *scalable* monitoring techniques to detect service violations, bandwidth theft, and attacks. The monitor alerts against possible DoS attacks in early stage
- Policy enforcement and controlling the suspected flows are needed to maintain *confidence* in the *security* and *QoS* of networks

Methods

- Network tomography
  - Stripe based probing is used to infer individual link loss from edge-to-edge measurements
  - Overlay network is used to identify congested links by measuring loss of edge-to-edge paths
- Transport layer flow characteristics are used to protect critical packets of a flow
- Edge-to-edge mechanism is used to detect and control unresponsive flows

Monitoring Network Domains

- Idea:
  - Excessive traffic changes internal characteristics inside a domain (high delay & loss, low throughput)
  - Monitor network domain for unusual patterns
  - If traffic is aggregating towards a domain (same IP prefix), probably an attack is coming
- Measure delay, link loss, and throughput achieved by user inside a network domain

Monitoring by periodic polling or deploying agents in high speed core routers put non-trivial overhead on them

Core-assisted loss measurements

- Core reports to the monitor whenever packet drop exceeds a local threshold
- Monitor computes the total drop for time interval t
- If the total drop exceeds a global threshold

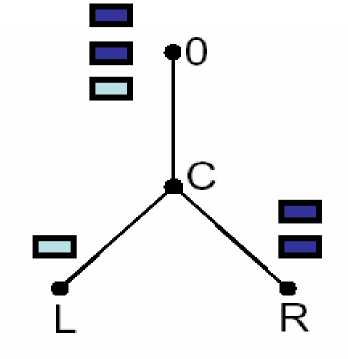
  a. The monitor sends a query to all edge routers requesting their current rates

  b. The monitor computes total incoming rate from all edge

  c. The monitor computes the loss ratio as the ratio of the dropped packets and the total incoming rate

  d. If the loss ratio exceeds the SLA loss ratio, a possible SLA violation is reported

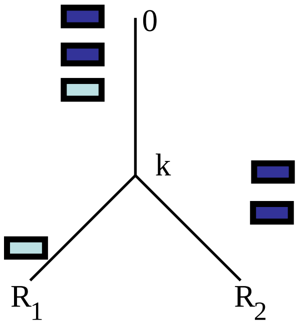
Stripe Unicast Probing [Duffield et al., *INFOCOM* ’01]



- Back-to-back packets experience similar congestion in a queue with a high probability
- Receiver observes the probes to correlate them for loss inference
- Infer internal characteristics using topology
- For general tree? Send stripe from root to every order-pair of leaves
- Develop stripe-based monitoring by extending loss inference for multiple drop precedence

Inferring Loss

- Calculate how many packets are received by the two receivers. Transmission probability A_k



$$A_k = \frac{Z_{R1}\, Z_{R2}}{Z_{R1 \cup R2}}$$

where Z_i binary variable which takes 1 when all packets reached their destination and 0 otherwise

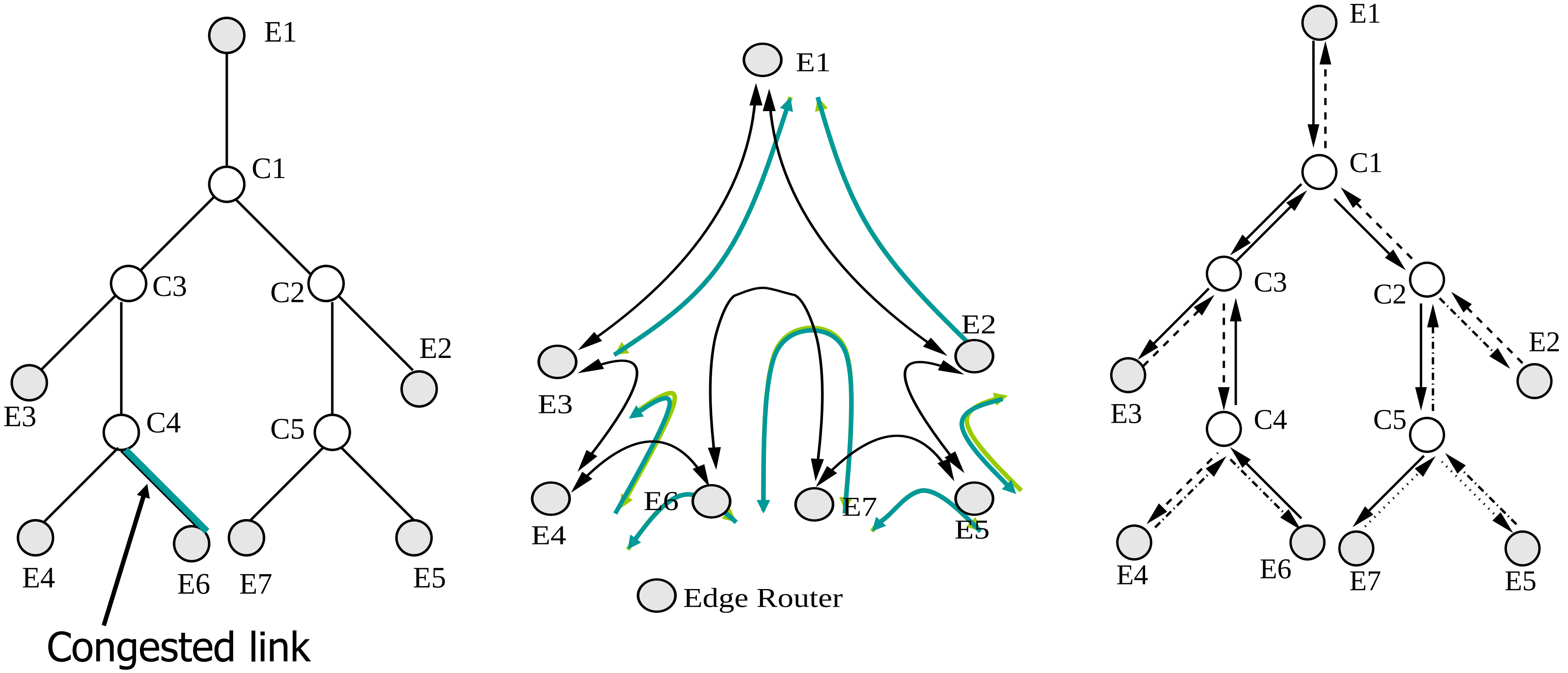
- Loss is 1 - A_k
- For general tree, send stripe from root to every order-pair of leaves.

Overlay-based Monitoring

- Problem statement
  - Given topology of a network domain, identify which links are congested
- Solutions: *Simple* and *Advanced* methods

1. Monitor the network for link delay
2. If $delay^i > Threshold^i_{delay}$ for path *i,* then probe the network for loss
3. If $loss^j > Threshold^j_{loss}$ for any link *j,* then probe the network for throughput
4. If $BW^k > Threshold^k_{BW}$, flow *k* is violating service agreements by taking excess resources. Upon detection, we control the flows.

Probing: Simple Method



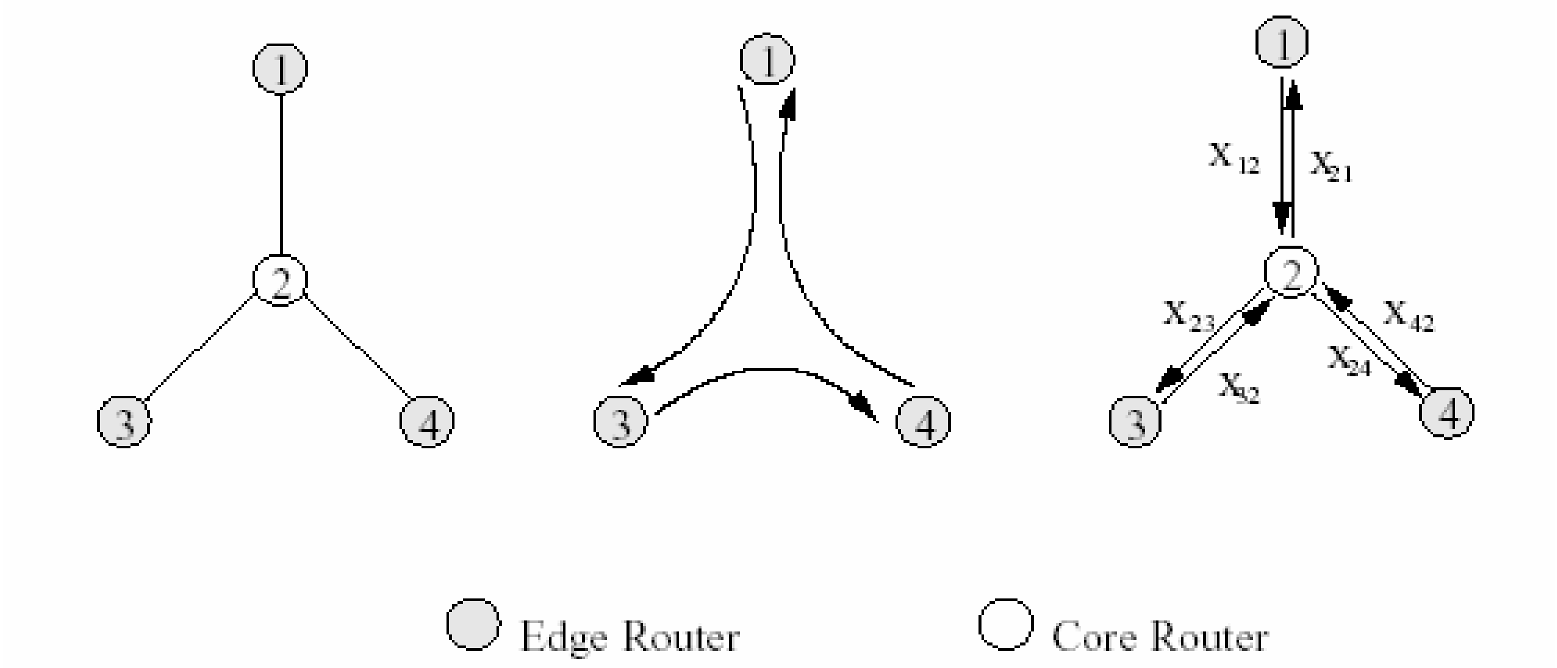
(a) Topology

(b) Overlay

(c) internal links

- Each peer probes both of its neighbors
- Detect congested link in both directions

An Example

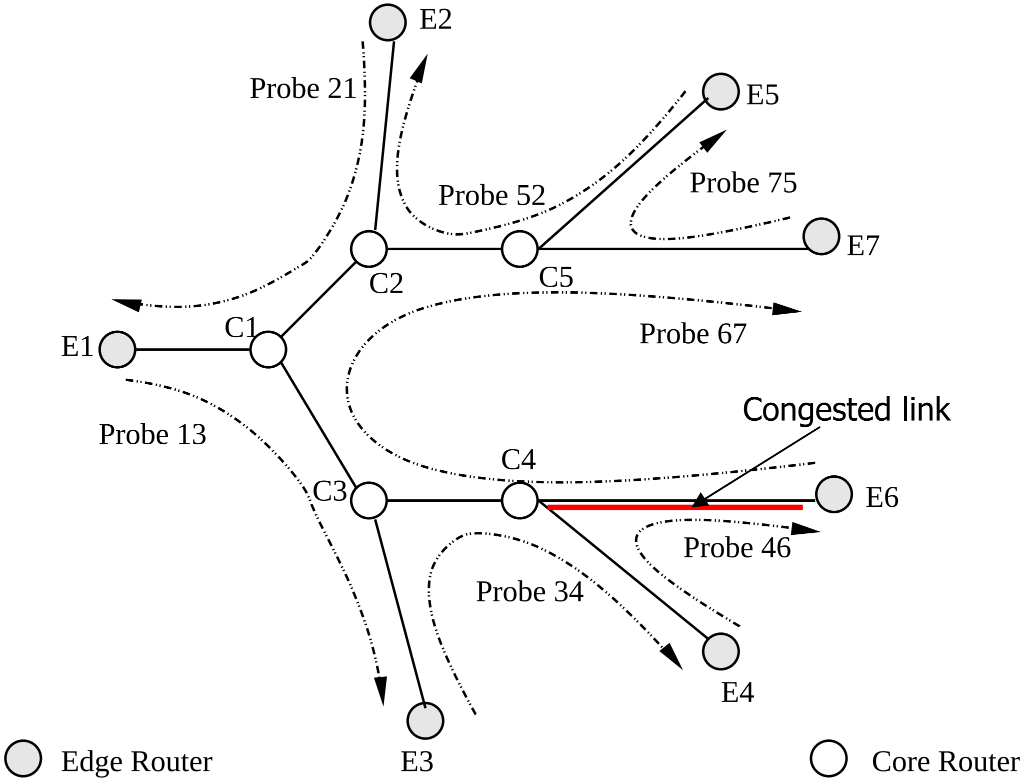


$$X_{1,2} + X_{2,3} = P_{1,3} \qquad X_{3,2} + X_{2,4} = P_{3,4} \qquad X_{4,2} + X_{2,1} = P_{4,1}$$

- Perform one round peer-to-peer probing in counter-clockwise direction
- Each boolean variable $\mathbf{X_{ij}}$ represents the congestion status of link **i → j**
- For each probe **P**, we have an equation $\mathbf{P_{i,j} = X_{i,k} + \ldots + X_{l,j}}$

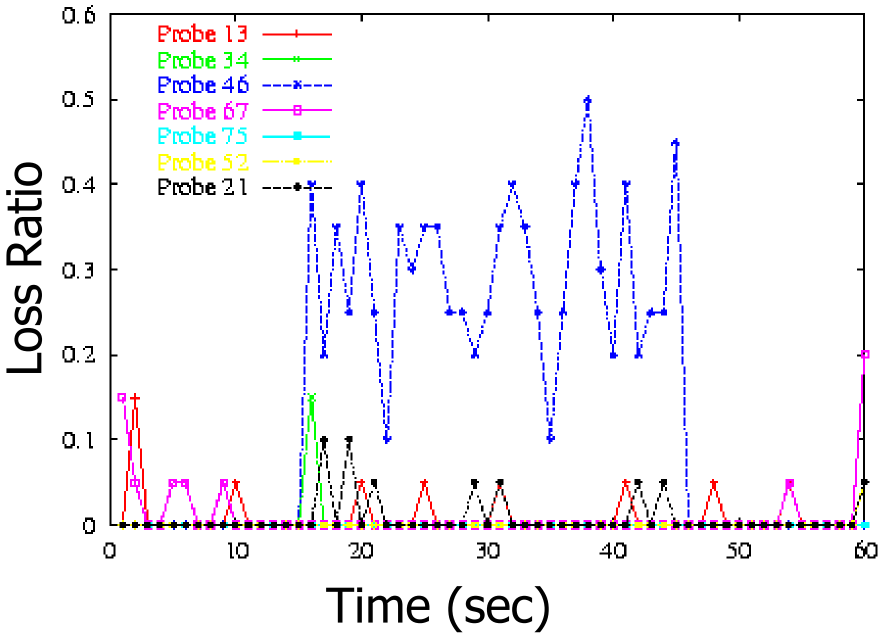
Experiments: Evaluation methodology

- Simulation using *ns-2*
- Two topologies
  - C-C links, 20 Mbps
  - E-C links, 10 Mbps
- Parameters
  - Number of flows order of thousands
  - Change life time of flows
  - Simulate attacks by varying traffic intensities and injecting traffic from multiple entry points
- Output Parameters
  - delay, loss ratio, throughput

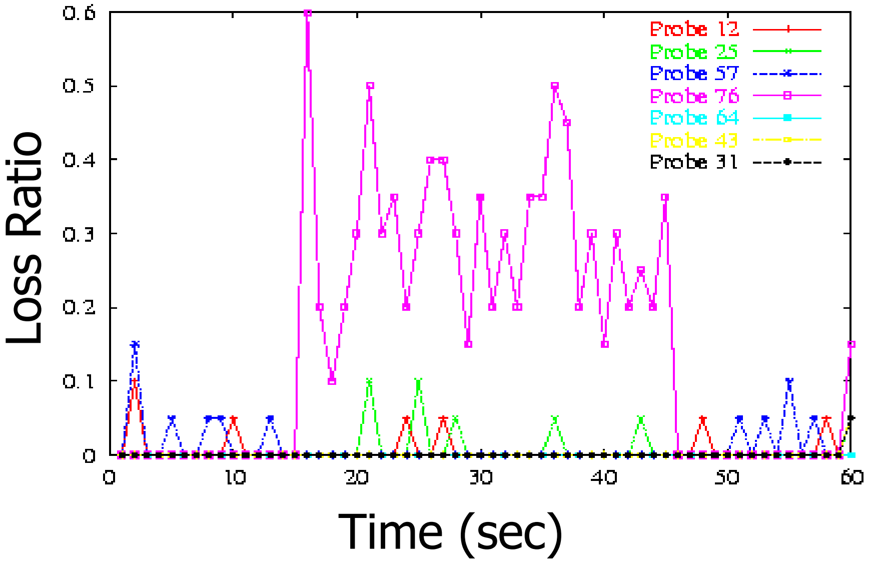


Topology 1

Identified Congested Links



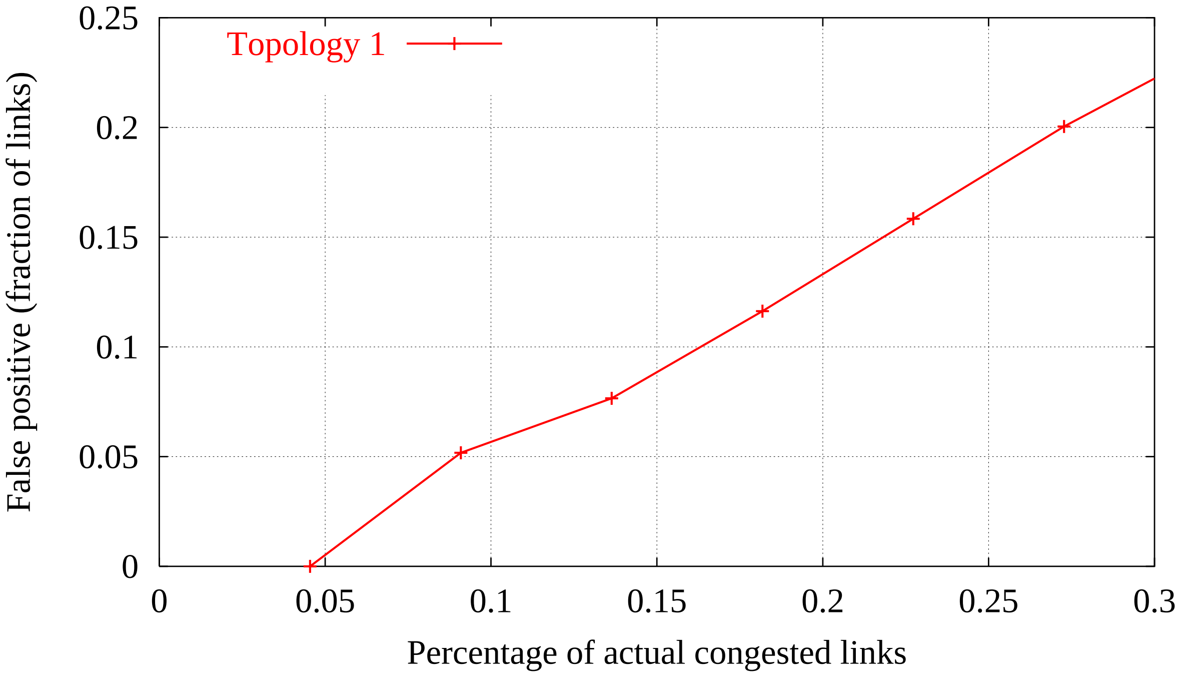
(a) Counter clockwise probing



(b) Clockwise probing

Probe46 in graph (a) and Probe76 in graph (b) observe high losses, which means link C4 → E6 is congested.

False Positive (theoretical analysis)



- The simple method does not correctly label all links
- The unsolved “good” links are considered bad hence false positive happens
- Need to refine the solution → Advanced Method

- Example:

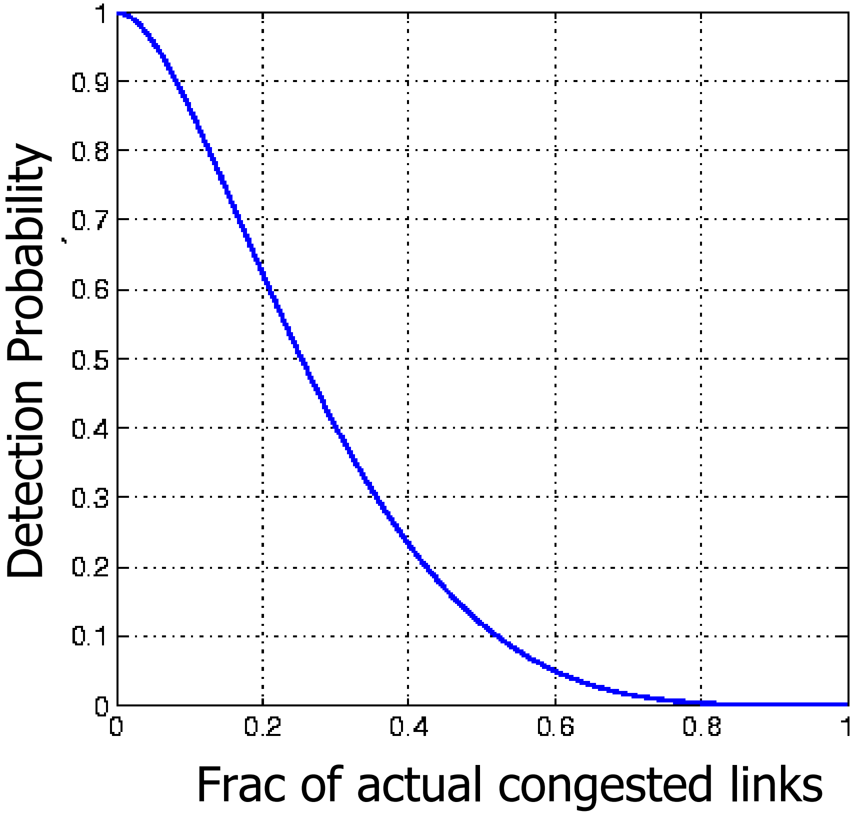
  if 100 links in the network and 20 of them are congested and 80 are “good”. The basic probing method can identify 15 congestion links and 70 good links. The other 15 are labeled as “unknown”. If all unknown links are treated as congested, 10 good link will be falsely labeled as congested. When the false positive is too high, the available paths that can be chosen by the routers are restricted, thus network performance is impacted.

Analyzing Simple Method

- Lemma 1. If *P* and *P'* are probe paths in the first and the second round of probing respectively, $|P \cap P'| \leq 1$
- Theorem 1. If only one probe path *P* is shown to be congested in any round of probing, the simple method successfully identifies status of each link in *P*
- Performs better if edge-to-edge paths are congested
- The average length of the probe paths in the Simple method is ≤ 4

Performance: Simple Method

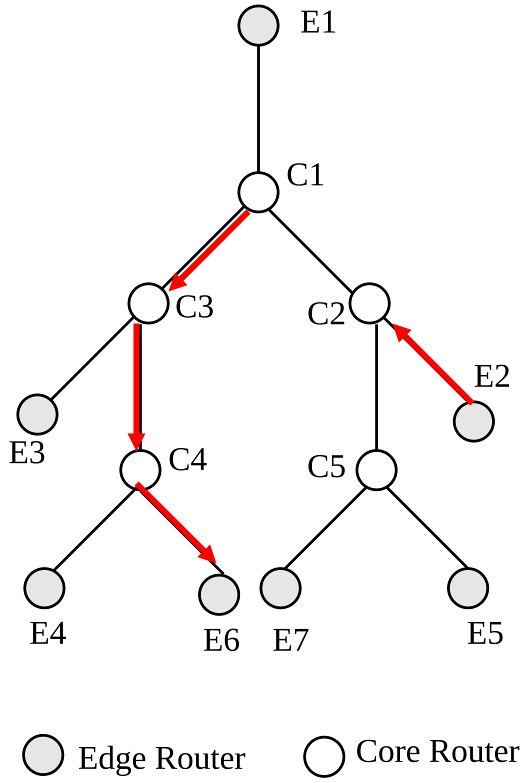
Theorem 2. Let *p* be the probability of a link being congested in any arbitrary overlay network. The simple method determines the status of any link of the topology with probability at least $2(1-p)^4-(1-p)^7+p(1-p)^{12}$

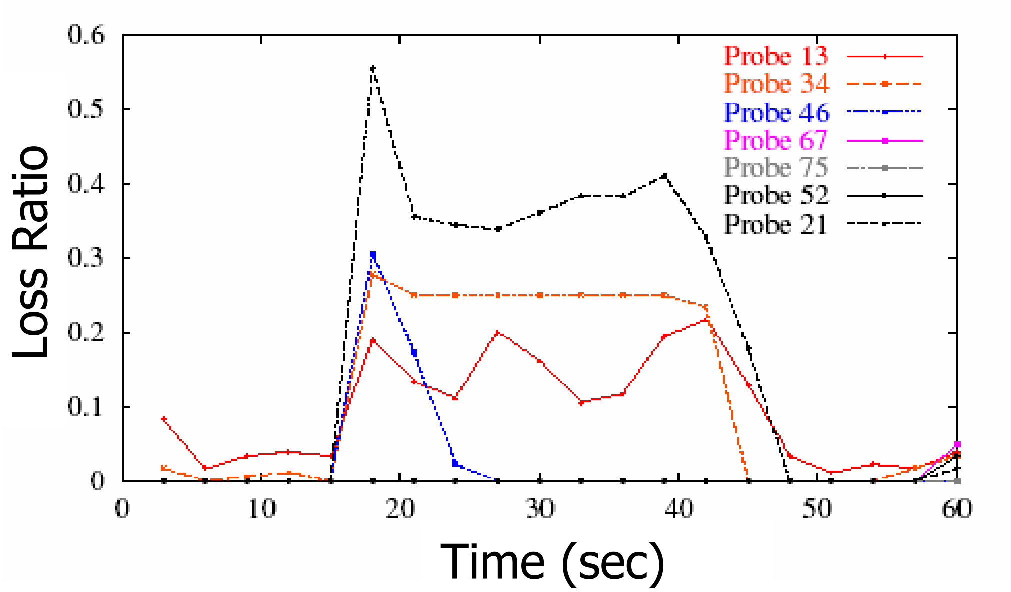


Advanced Method

```
AdvancedMethod()
begin
  Conduct Simple Method. E is the unsolved equation set
  for Each undecided variable $X_{ij}$ of E do
    node1 = FindNode(Tree T, $v_i$, IN)
    node2 = FindNode(Tree T, $v_j$ , OUT)
    if node1 ≠ NULL AND node2 ≠ NULL then
      Probe(node1, node2). Update equation set E
    end if
    Stop if no more probe exists
  endfor
end
```

Identifying Links: Advanced Method





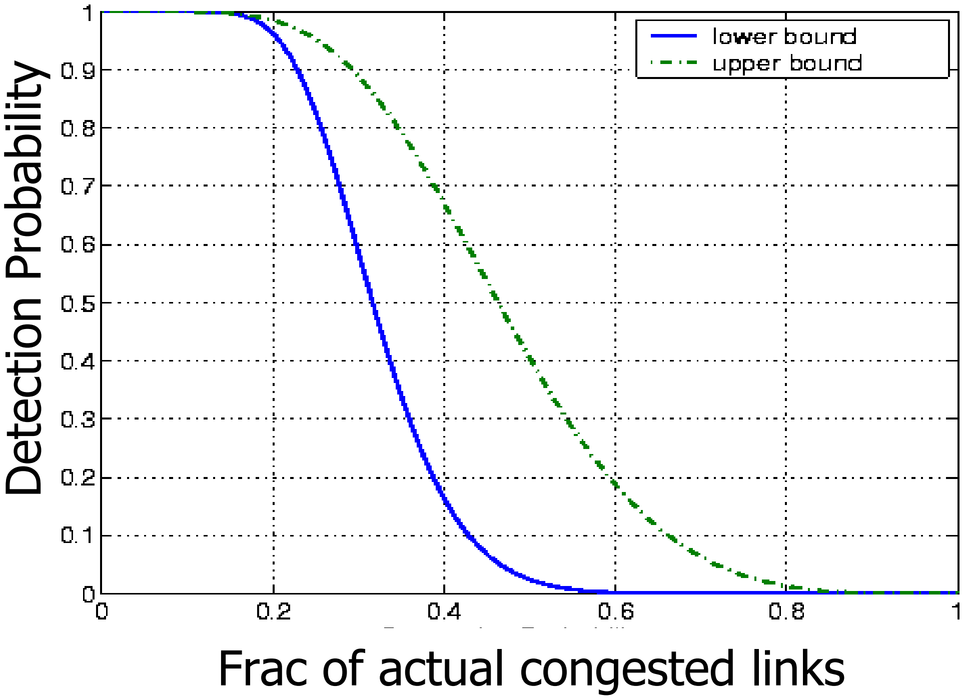
Link E2 → C2, C1 → C3, C3 → C4, and C4 → E6 are congested. Simple method identifies all except E2 → C2. Advanced method finds probe E5→E1 to identify status of E2 → C2.

Analyzing Advanced Method

- Lemma 2. For an arbitrary overlay network with *n* edge routers, on the average a link lies on *b* = $\frac{n(3n-2)}{8\log n}$ edge-to-edge paths
- Lemma 3. For an arbitrary overlay network with *n* edge routers, the average length of all edge-to-edge paths is *d* = $\frac{3n}{2\log n}$
- Theorem 3. Let *p* be the probability of a link being congested. The advanced method can detect the status of a link with probability at least (1-(1-(1-p)^{d})^{b})

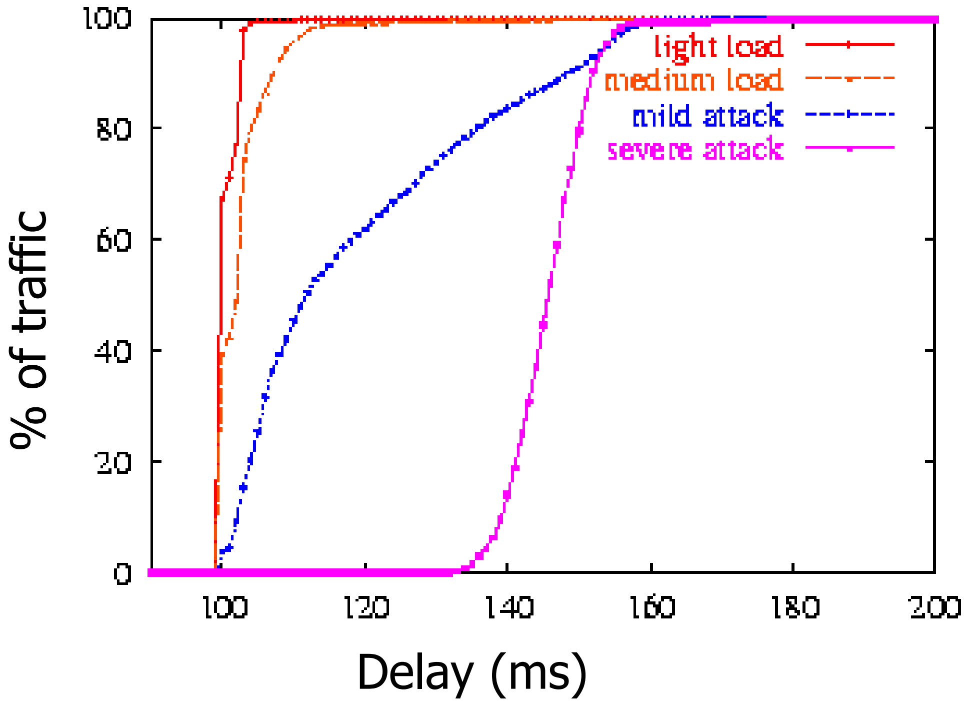
Bounds on Advanced Method

- Graph shows lower and upper bounds
- When congestion is ≤ 20%, links are identified with *O(n)* probes with probability ≥ 0.98
- Does not help if ≥ 60% links are congested



Advanced method uses output of simple method and topology to find a probe that can be used to identify status of an unsolved link in simple method

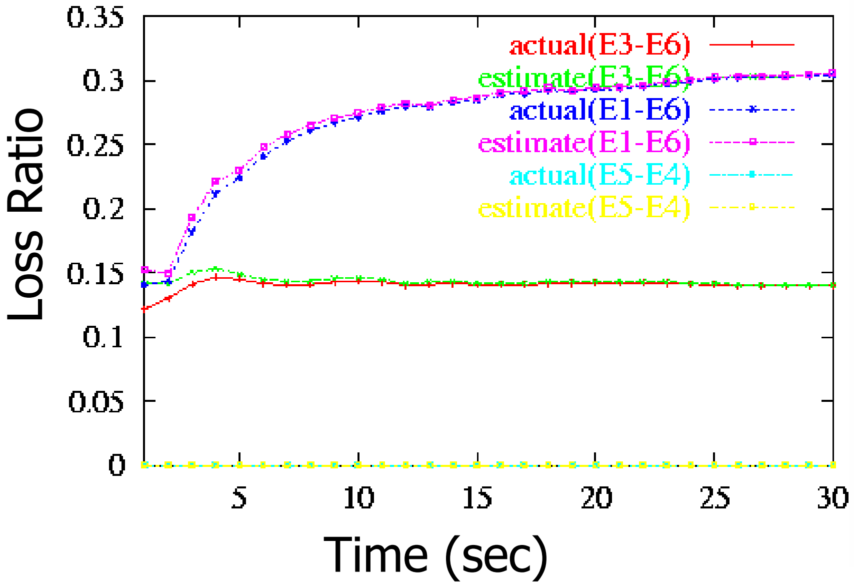
Experiments: Delay Measurements



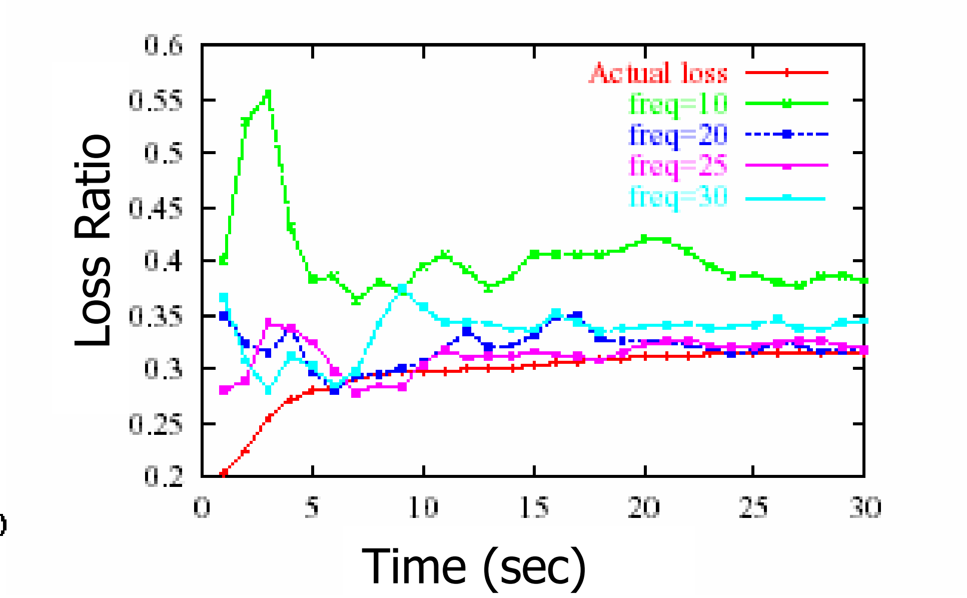
Cumulative distribution function (cdf)

- Attack changes delay pattern in a network domain
- We need to know the delay pattern when there is not attack

Experiments: Loss measurements



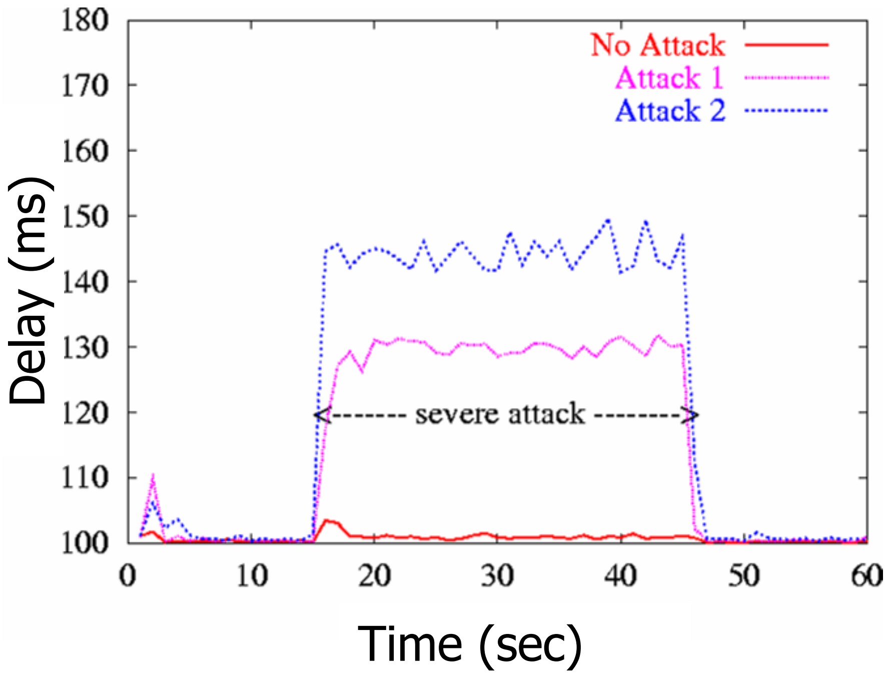
(a) Core-assisted



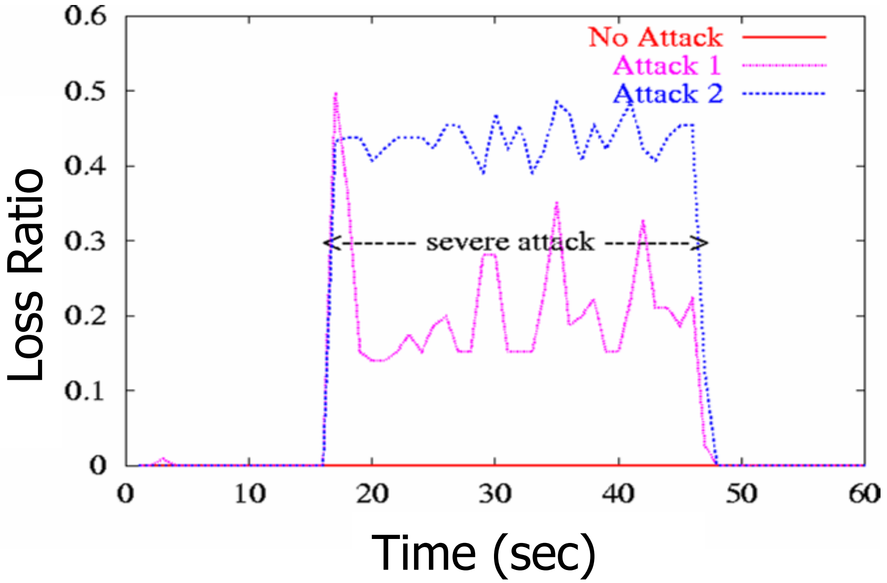
(b) Stripe-based

Core-based measurement is more precise than stripe-based, however, it has high overhead

Attack Scenarios



(a) Changing delay pattern due to attack



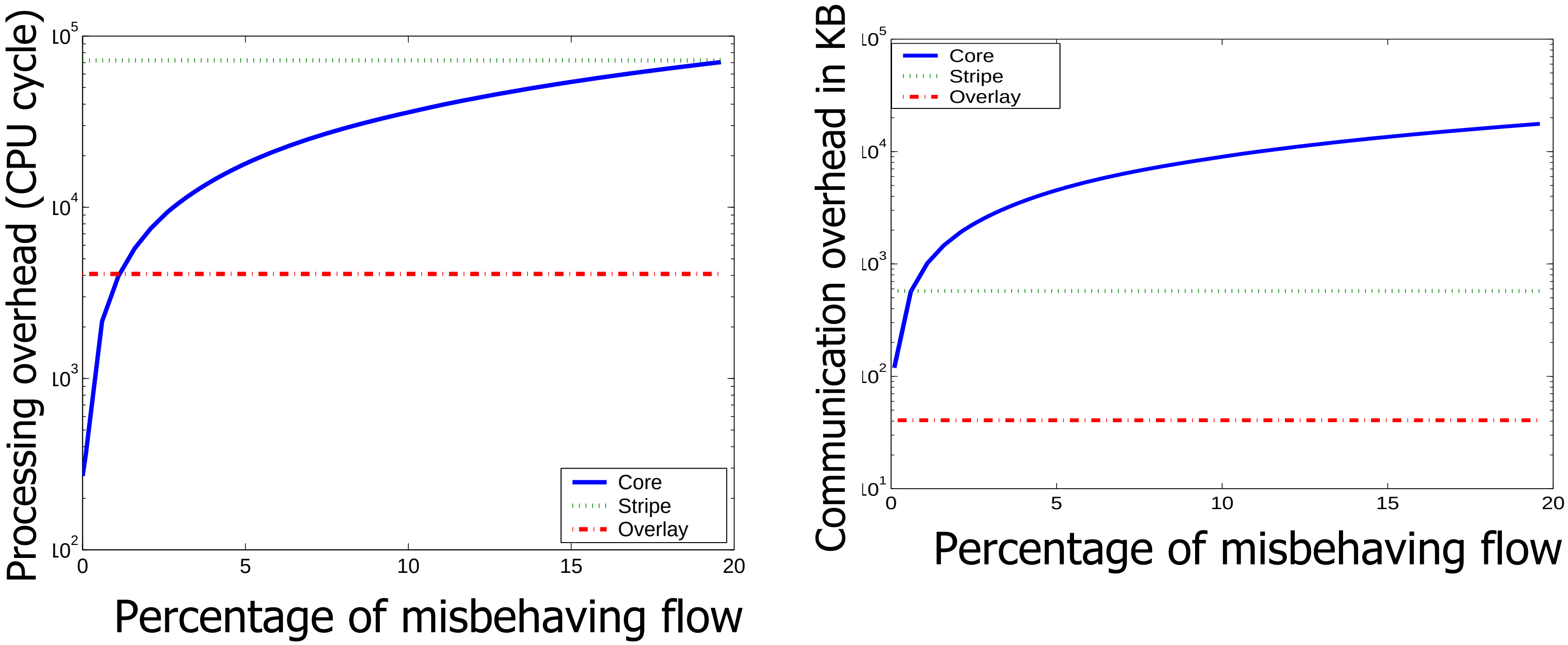
(b) Changing loss pattern due to attack

- Attack 1 violates SLA and causes 15-30% of packet loss
- Attack 2 causes more than 35% of packet loss

Detecting DoS Attacks

- If many flows aggregate towards a downstream domain, it might be a DoS attack on the domain
- Analyze flows at exit routers of the congested links to identify misbehaving flows
- Activate filters to control the suspected flows
- Flow association with ingress routers
  - Egress routers can backtrack paths, and confirm entry points of suspected flows

Overhead comparison



(a) Processing overhead

(b) Communication overhead

- Core has relative low processing overhead
- Overlay scheme has an edge over other two schemes

Observations

- Stripe-based Monitoring
  - Stripe-based probing can monitor DiffServ networks only from the edges
  - It takes 10 sec to converge the inferred loss ratio to actual loss ratio with ≥ 90% accuracy
  - 10-15 delay probes and 20-25 loss probes per second are sufficient for monitoring
  - Probe is a 3-packet stripe
    - 3 shows good correlation, 4 does not add much

Observations (Cont’d)

- Overlay-based Monitoring
  - – Congestion status of individual links can be inferred from edge-to-edge measurements
  - – When the network is ≤ 20% congested
    - Status of a link is identified with probability ≥ 0.98
    - Requires *O(n)* probes, where *n* is the number of edge routers
  - – Worst case is $O(n^2)$, whereas stripe-based requires $O(n^3)$ probes to achieve same functionality

Observations (Cont’d)

- Analyze existing techniques to defeat DoS attacks
  - Marking has less overhead than Filtering, however, it is only a forensic method
  - Monitoring might have less processing overhead than marking or filtering, however, monitoring injects packets and others do not
  - Monitoring can alert against DoS attacks in early stage

Observations (Cont’d)

- Traffic Conditioner
  - Using small state table, we can design scalable traffic conditioner
  - It can protect critical packets of a flow to improve application QoS (delay, throughput, response time, …)
  - Both Round trip time (RTT) & Retransmission time-out (RTO) are necessary to avoid RTT-bias among flows

Observations (Cont’d)

- Flow Control
  - Network tomography is used to design edge-to-edge mechanism to detect & control unresponsive flows
  - QoS of adaptive flows improves significantly with flow control mechanism

Conclusion on Monitoring

- Elegant way to use probability in inferring loss. 3-packets stripe shows good correlation
- Monitoring network can detect service violation and bandwidth theft using measurements
- Monitoring can detect DoS attacks in early stage. Filter can be used to stop the attacks
- Overlay-based monitoring requires only *O(n)* probing with a very high probability, where *n* is the number of edge routers
- Overlay-based monitoring has very low communication and processing overhead
- Stripe-based inference is useful to annotate a topology tree with loss, delay, and bandwidth.

D. Intruder Identification in Ad Hoc Networks

- Problem Statement

  Intruder identification in ad hoc networks is the procedure of identifying the user or host that conducts the inappropriate, incorrect, or anomalous activities that threaten the connectivity or reliability of the networks and the authenticity of the data traffic in the networks

Research Motivation

- More than ten routing protocols for Ad Hoc networks have been proposed
- Research focuses on performance comparison and optimizations such as multicast and multiple path detection
- Research is needed on the security of Ad Hoc networks.
- Applications: Battlefields, disaster recovery.

Research Motivation

- Two kinds of attacks target Ad Hoc network
  - External attacks:
    - MAC Layer jam
    - Traffic analysis
  - Internal attacks:
    - Compromised host sending false routing information
    - Fake authentication and authorization
    - Traffic flooding

Research Motivation

- Protection of Ad Hoc networks
  - Intrusion Prevention
    - Traffic encryption
    - Sending data through multiple paths
    - Authentication and authorization
  - Intrusion Detection
    - Anomaly pattern examination
    - Protocol analysis study

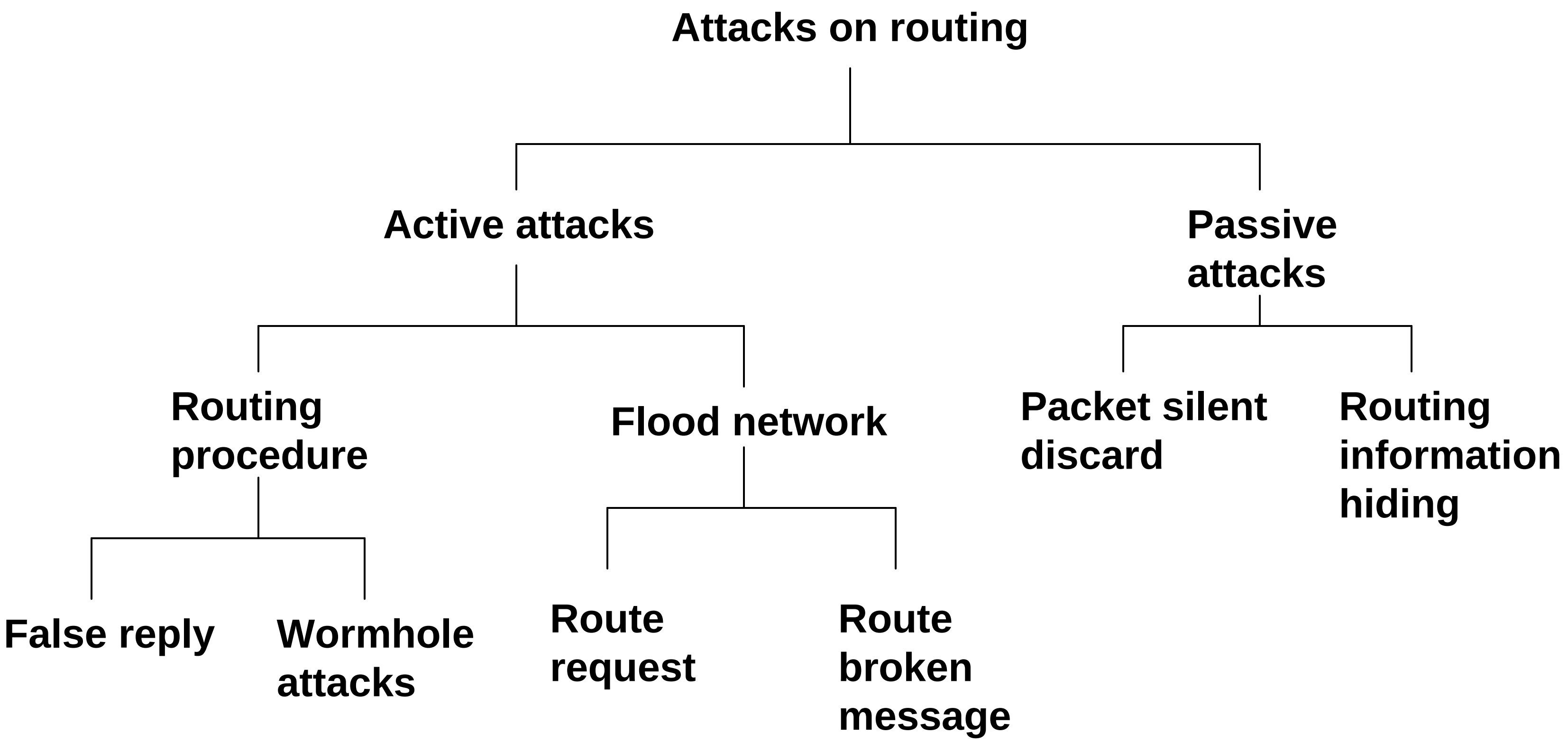
Research Motivation

- Deficiency of intrusion prevention
  - – increase the overhead during normal operation period of Ad Hoc networks
  - – The restriction on power consumption and computation capability prevent the usage of complex encryption algorithms
  - – Flat infrastructure increases the difficulty for the key management and distribution
  - – Cannot guard against internal attacks

Research Motivation

- Why intrusion detection itself is not enough
  - Detecting intrusion without isolating the malicious host leaves the protection in a passive mode
  - Identifying the source of the attack may accelerate the detection of other attacks

Attacks on routing in mobile ad hoc networks



Attacks on routing

Active attacks

Passive attacks

Routing procedure

Flood network

Packet silent discard

Routing information hiding

False reply

Wormhole attacks

Route request

Route broken message

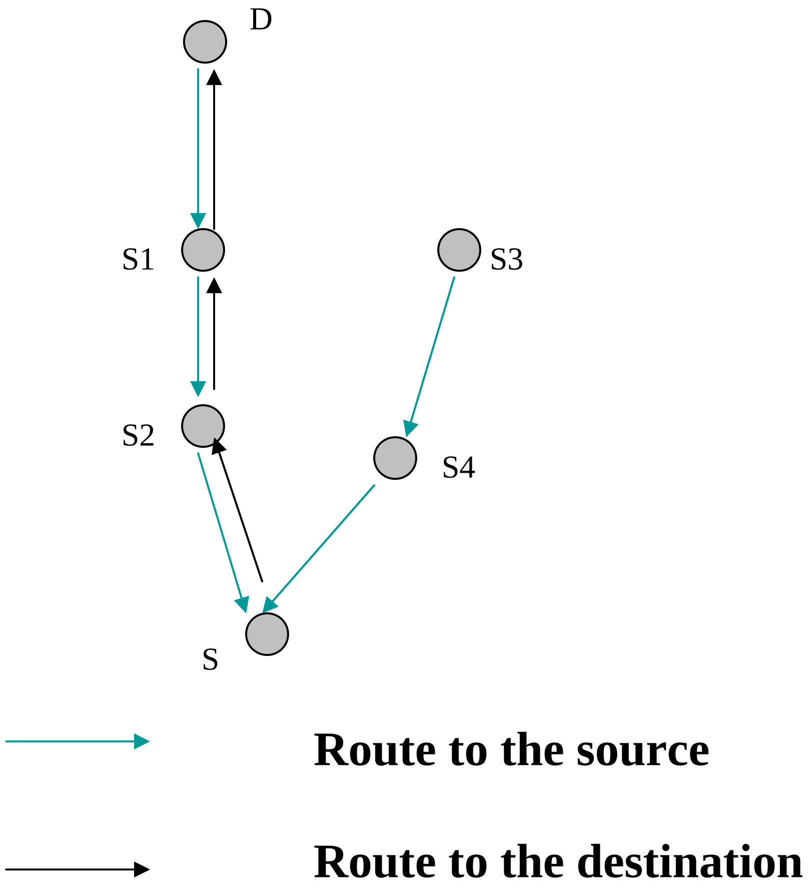
Ideas

- Monitor the sequence numbers in the route request packets to detect abnormal conditions
- Apply reverse labeling restriction to identify and isolate attackers
- Combine local decisions with knowledge from other hosts to achieve consistent conclusions
- Combine with trust assessment methods to improve robustness

Introduction to AODV

- Introduced in 97 by Perkins at NOKIA, Royer at UCSB
- 12 versions of IETF draft in 4 years, 4 academic implementations, 2 simulations
- Combines on-demand and distance vector
- Broadcast Route Query, Unicast Route Reply
- Quick adaptation to dynamic link condition and scalability to large scale network
- Support multicast

Route Discovery in AODV (An Example)



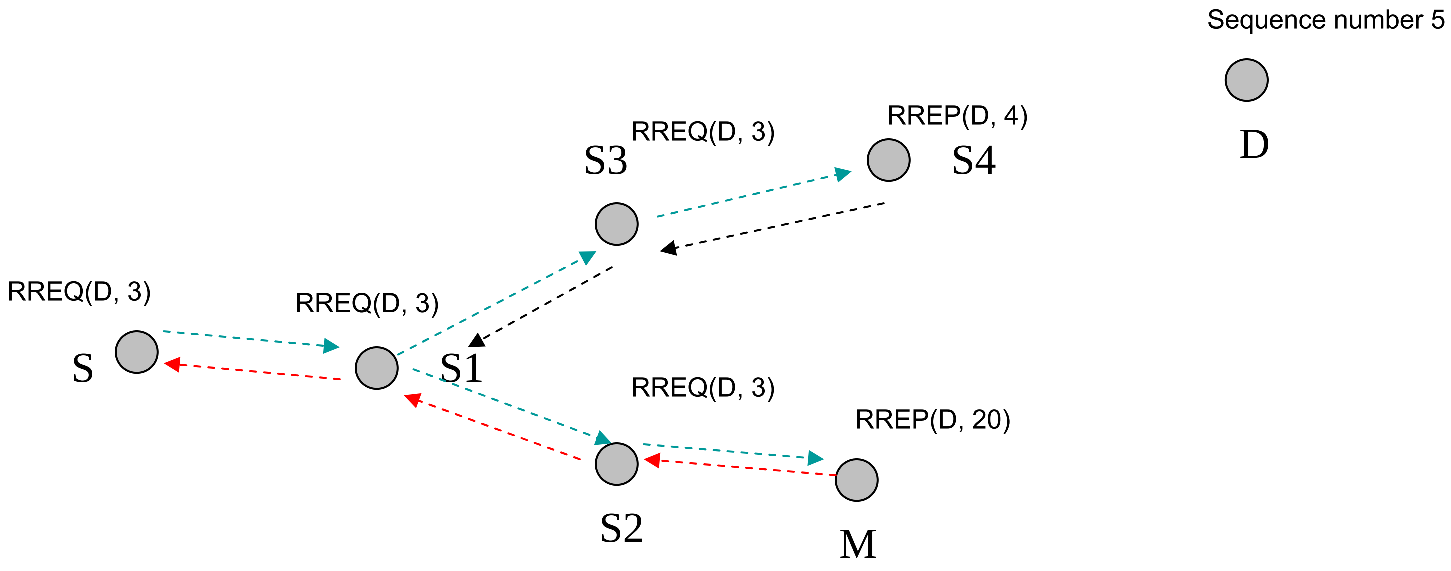
Route to the source

Route to the destination

Attacks on AODV

- Route request flooding
  - query non-existing host (RREQ will flood throughout the network)
- False distance vector
  - reply “one hop to destination” to every request and select a large enough sequence number
- False destination sequence number
  - select a large number (even beat the reply from the real destination)
- Wormhole attacks
  - tunnel route request through wormhole and attract the data traffic to the wormhole
- Coordinated attacks
  - The malicious hosts establish trust to frame other hosts, or conduct attacks alternatively to avoid being identified

False Destination Sequence Attack

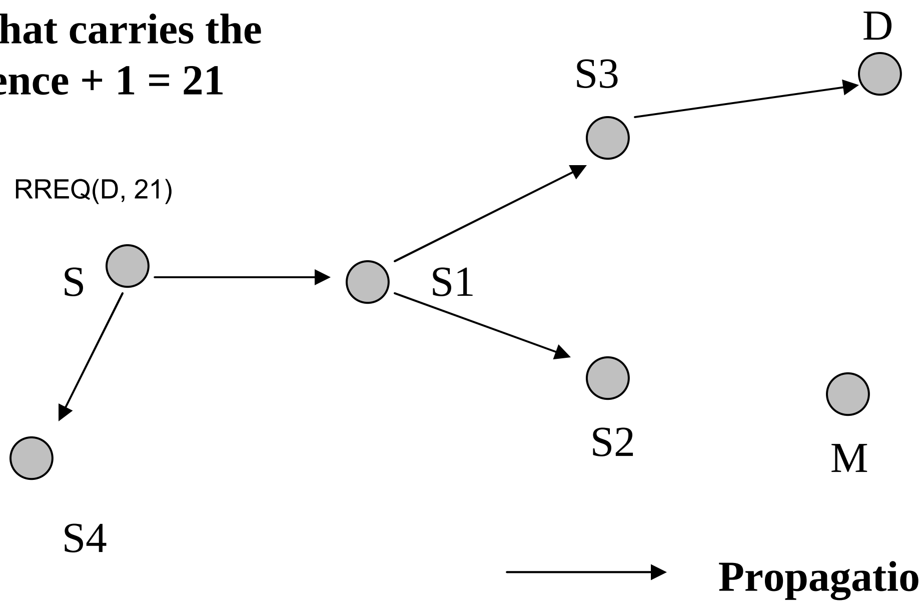


Packets from S to D are sinking at M.

During Route Rediscovery, False Destination Sequence Number Attack Is Detected, S needs to find D again.

Node movement breaks the path from S to M (trigger route rediscovery).

(1). S broadcasts a request that carries the old sequence + 1 = 21

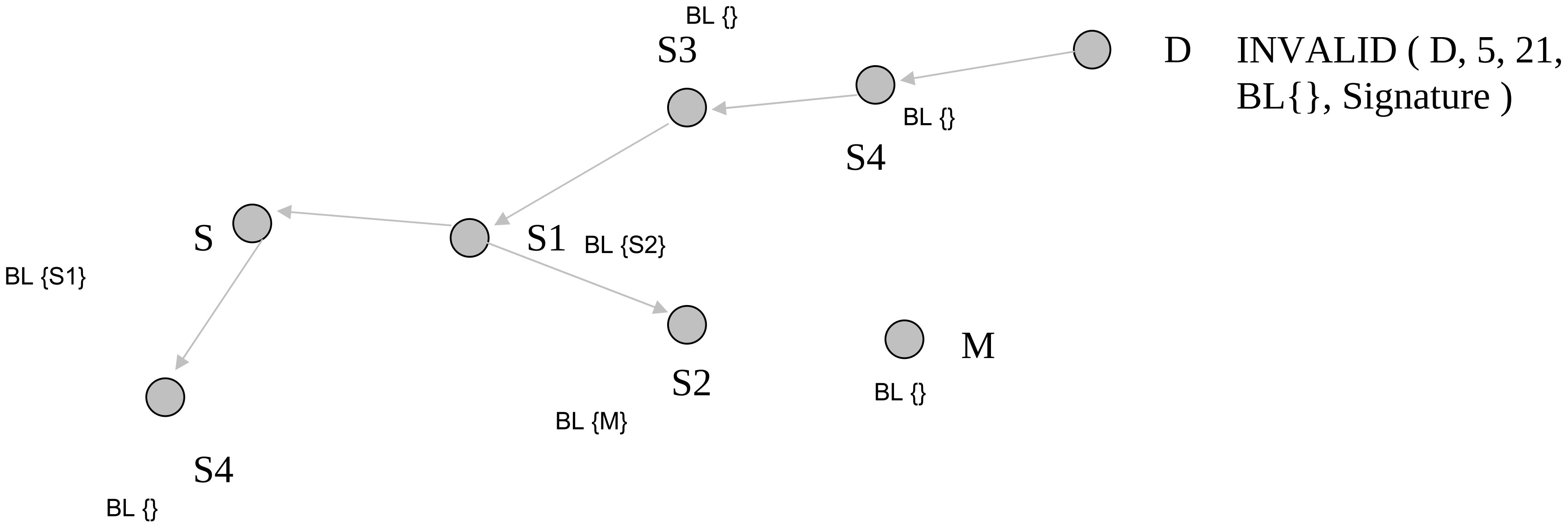


(2) D receives the RREQ. Local sequence is 5, but the sequence in RREQ is 21. D detects the false destination sequence number attack.

Reverse Labeling Restriction (RLR)

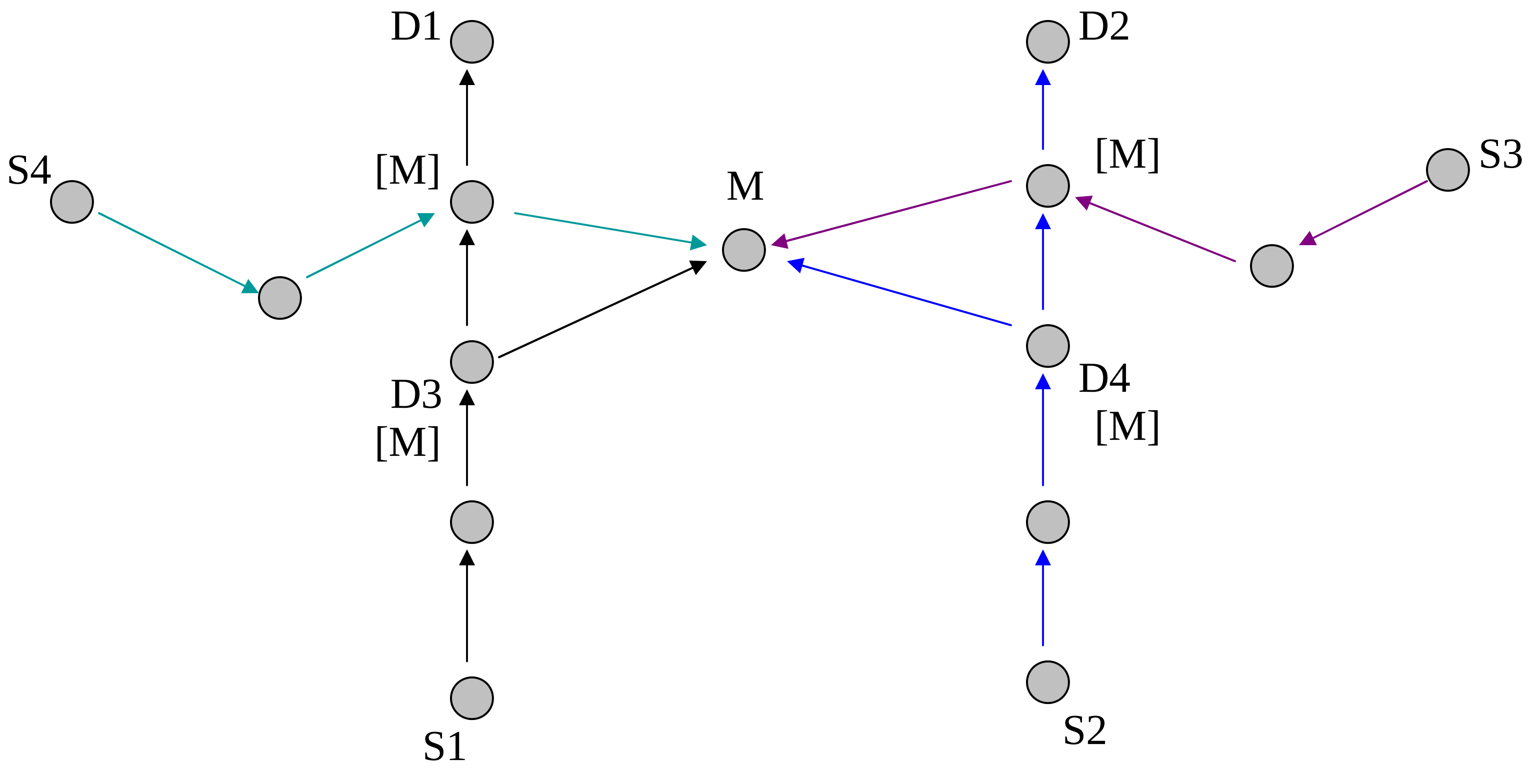
Blacklists are updated after an attack is detected.

- Basic Ideas
  - Every host maintains a blacklist to record suspicious hosts who gave wrong route related information.
  - The destination host will broadcast an INVALID packet with its signature. The packet carries the host’s identification, current sequence, new sequence, and its own blacklist.
  - Every host receiving this packet will examine its route entry to the destination host. The previous host that provides the false route will be added into this host’s blacklist.



Correct destination sequence number is broadcasted.

Blacklist at each host in the path is determined.

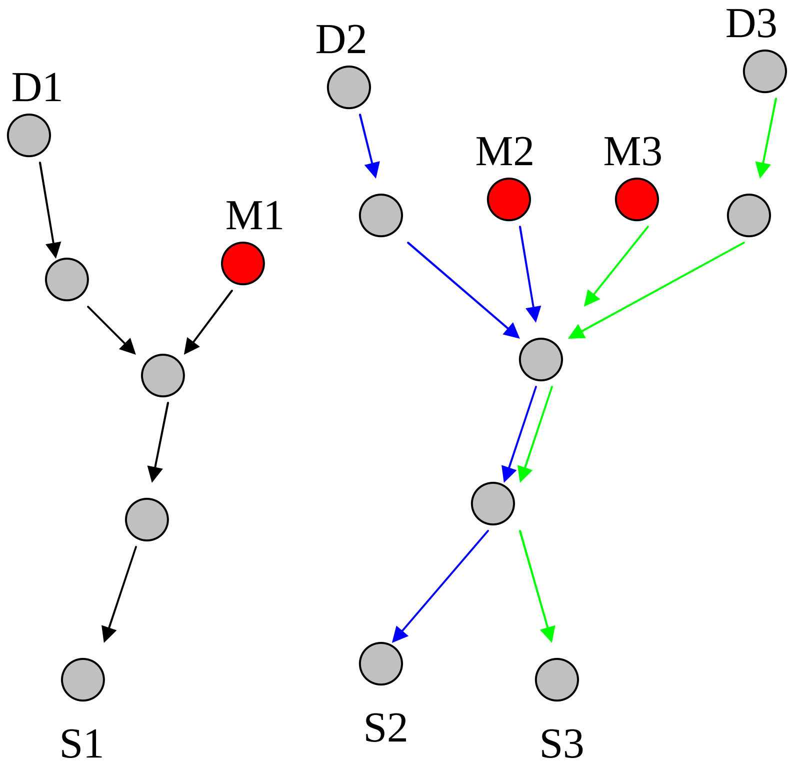


M attacks 4 routes (S1-D1, S2-D2, S3-D3, and S4-D4). When the first two false routes are detected, D3 and D4 add M into their blacklists. When later D3 and D4 become victim destinations, they will broadcast their blacklists, and every host will get two votes that M is malicious host.

Malicious site is in blacklists of multiple destination hosts.

- If M is in multiple blacklists, M is classified as a malicious host based on a certain threshold.
- Intruder is approximately identified.
- Trust values can be used for combining knowledge from other hosts.

Acceleration in Intruder Identification



Coordinated attacks by M1, M2, and M3

Multiple attackers trigger more blacklists to be broadcasted by D1, D2, D3.

Reverse Labeling Restriction (RLR)

- Update Blacklist by Broadcasted Packets from Destinations under Attack
  - Next hop on the false route will be put into local blacklist, and a counter increases. The time duration that the host stays in blacklist increases exponentially to the counter value.
  - When timer expires, the suspicious host will be released from the blacklist and routing information from it will be accepted.

Deal With Hosts in Blacklist

- Packets from hosts in blacklist
  - Route request: If the request is from suspicious hosts, ignore it.
  - Route reply: If the previous hop is suspicious and the query destination is not the previous hop, the reply will be ignored.
  - Route error: Will be processed as usual. RERR will activate re-discovery, which will help to detect attacks on destination sequence.
  - Broadcast of INVALID packet: If the sender is suspicious, the packet will be processed but the blacklist will be ignored.

Attacks of Malicious Hosts on RLR

- Attack 1: Malicious host M sends false INVALID packet
  - Because the INVALID packets are signed, it cannot send the packets in other hosts' name
  - If M sends INVALID in its own name
    - If the reported sequence number is greater than the real sequence number, every host ignores this attack
    - If the reported sequence number is less than the real sequence number, RLR will converge at the malicious host. M is included in blacklist of more hosts. M accelerated the intruder identification directing towards M.

- Attack 2: Malicious host M frames other innocent hosts by sending false blacklist
  - If the malicious host has been identified, the blacklist will be ignored
  - If the malicious host has not been identified, this operation can only make the threshold lower. If the threshold is selected properly, it will not impact the identification results.
  - Combining trust can further limit the impact of this attack.

- Attack 3: Malicious host M only sends false destination sequence about some special host
  - The special host will detect the attack and send INVALID packets.
  - Other hosts can establish new routes to the destination by receiving the INVALID packets.

Experimental Studies of RLR

- The experiments are conducted using ns2.
- Various network scenarios are formed by varying the number of independent attackers, number of connections, and host mobility.
- The examined parameters include:
  - Packet delivery ratio
  - Identification accuracy: false positive and false negative ratio
  - Communication and computation overhead

Simulation Parameter

Simulation duration	1000 seconds
Simulation area	1000 * 1000 m
Number of mobile hosts	30
Transmission range	250 m
Pause time between the host reaches current target and moves to next target	0 – 60 seconds
Maximum speed	5 m/s
Number of CBR connection	25/50
Packet rate	2 pkt / sec

Experiment 1: Measure the Changes in Packet Delivery Ratio

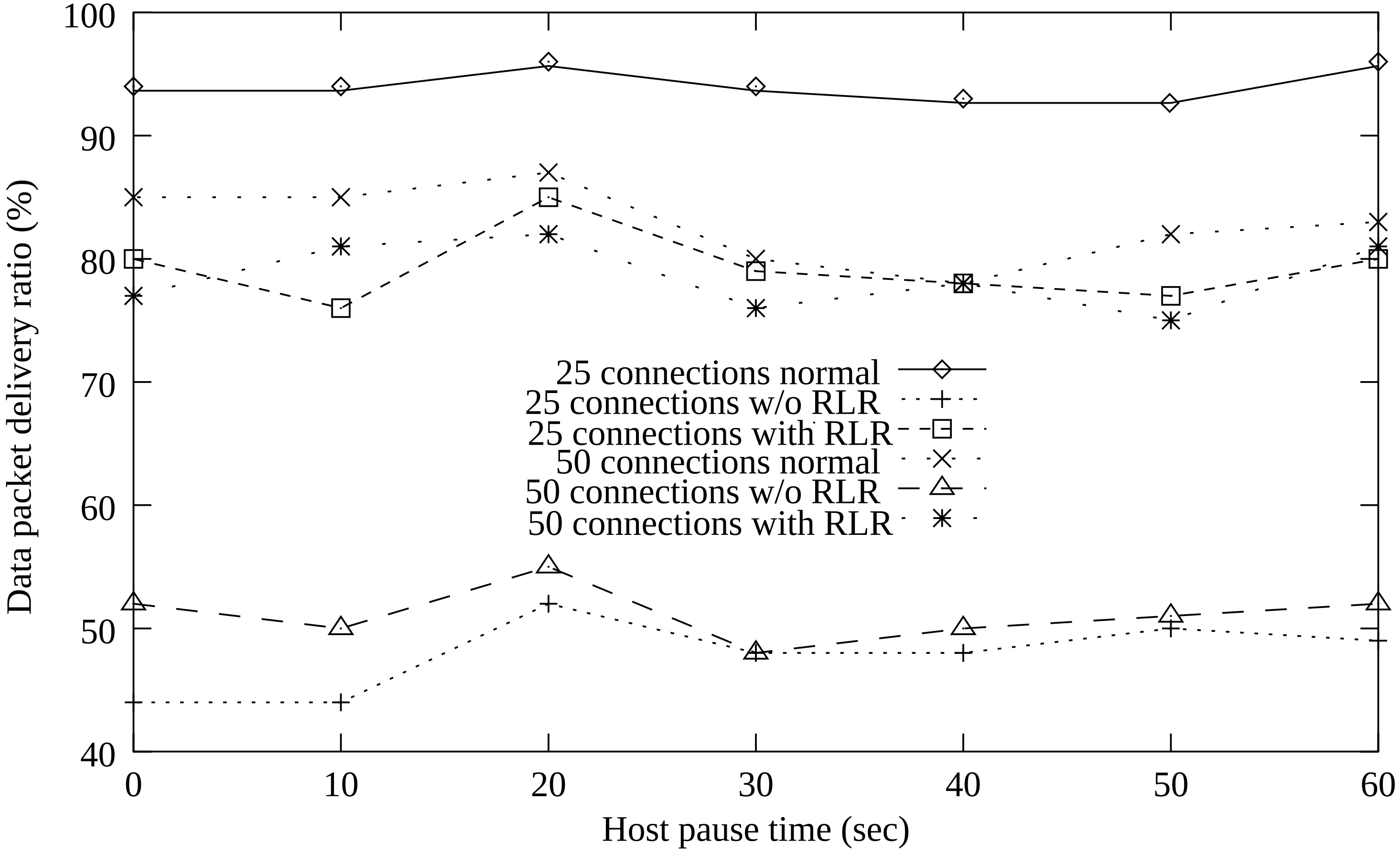
Purpose: investigate the impacts of host mobility, number of attackers, and number of connections on the performance improvement brought by RLR

Input parameters: host pause time, number of independent attackers, number of connections

Output parameters: packet delivery ratio

Observation: When only one attacker exists in the network, RLR brings a 30% increase in the packet delivery ratio. When multiple attacker exist in the system, the delivery ratio will not recover before all attackers are identified.

Increase in Packet Delivery Ratio: Single Attacker



X-axis is host pause time, which evaluates the mobility of host. Y-axis is delivery ratio. 25 connections and 50 connections are considered. RLR brings a 30% increase in delivery ratio. 100% delivery is difficult to achieve due to network partition, route discovery delay and buffer.

Experiment 2: Measure the Accuracy of Intruder Identification

Purpose: investigate the impacts of host mobility, number of attackers ,and connection scenarios on the detection accuracy of RLR

Input parameters: number of independent attackers, number of connections, host pause time

Output parameters: false positive alarm ratio, false negative alarm ratio

Observation: The increase in connections may improve the detection accuracy of RLR. When multiple attackers exist in the network, RLR has a high false positive ratio.

Accuracy of RLR: Single Attacker

	30 hosts, 25 connections		30 hosts, 50 connections	
Host Pause time (sec)	# of normal hosts identify the attacker	# of normal hosts marked as malicious	# of normal hosts identify the attacker	# of normal hosts marked as malicious
0	24	0.22	29	2.2
10	25	0	29	1.4
20	24	0	25	1.1
30	28	0	29	1.1
40	24	0	29	0.6
50	24	0.07	29	1.1
60	24	0.07	24	1.0

The accuracy of RLR when there is only one attacker in the system

Experiment 3: Measure the Communication Overhead

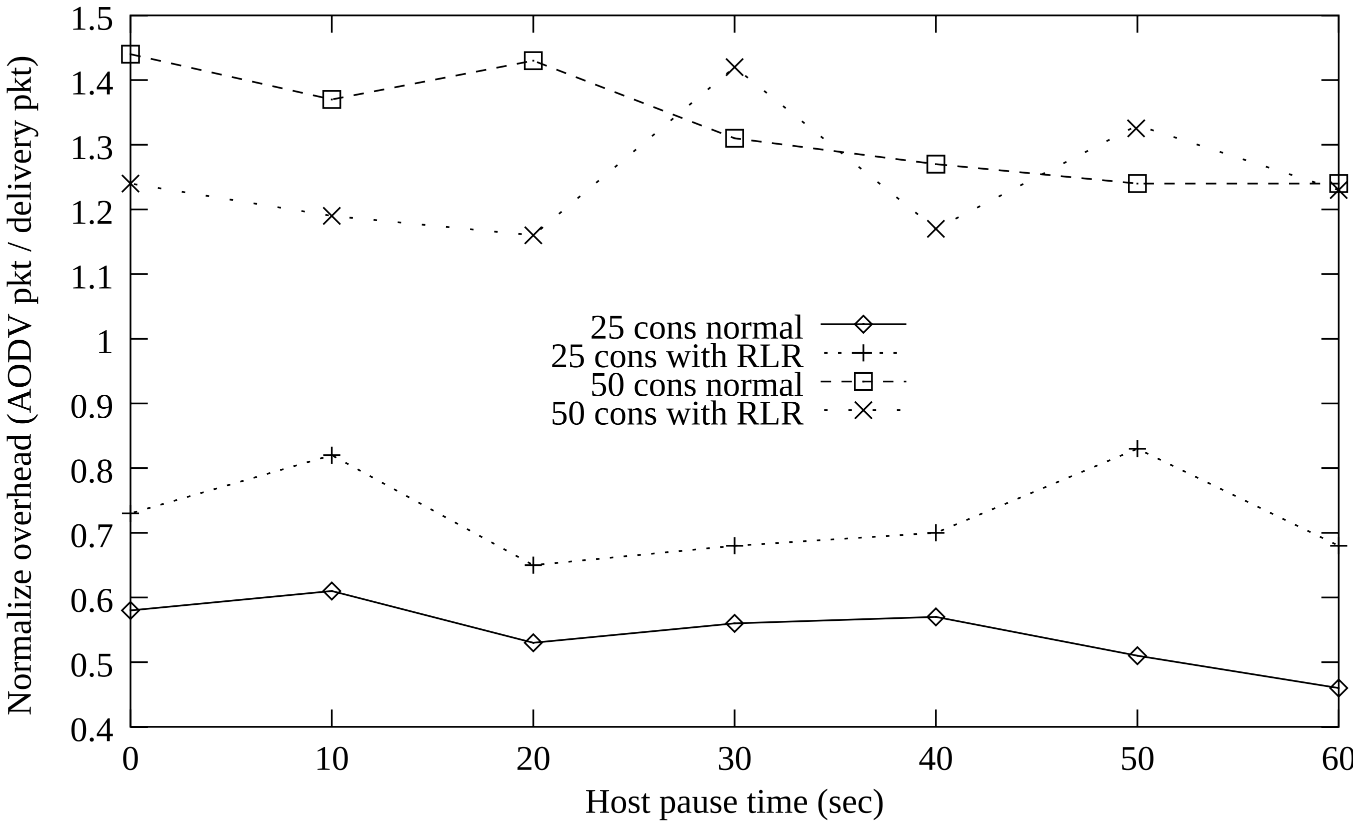
Purpose: investigate the impacts of host mobility and connection scenarios on the overhead of RLR

Input parameters: number of connections, host pause time

Output parameters: control packet overhead

Observation: When no false destination sequence attacks exist in the network, RLR introduces small packet overhead into the system.

Control Packet Overhead



X-axis is host pause time, which evaluates the mobility of host. Y-axis is normalized overhead (# of control packet / # of delivered data packet). 25 connections and 50 connections are considered. RLR increases the overhead slightly.

Research Opportunities: Improve Robustness of RLR

- Protect the good hosts from being framed by malicious hosts
  - The malicious hosts can frame the good hosts by putting them into blacklist.
  - By lowering the trust values of both complainer and complainee, we can restrict the impacts of the gossip distributed by the attackers.

- Avoid putting every host into blacklist
  - Combining the host density and movement model, we can estimate the time ratio that two hosts are neighbors
  - The counter for a suspicious host decreases as time passes
  - Adjusting the decreasing ratio to control the average percentage of time that a host stays in the blacklist of another host

- Defend against coordinated attacks
  - The behaviors of collusive attackers show Byzantine manners. The malicious hosts may establish trust to frame other hosts, or conduct attacks alternatively to avoid being identified.
  - Look for the effective methods to defend against such attacks. Possible research directions include:
    - Apply classification methods to detect the hosts that have similar behavior patterns
    - Study the behavior histories of the hosts that belong to the same group and detect the pattern of malicious behavior (time-based, order-based)

Conclusions on Intruder Identification

- False destination sequence attacks can be detected by the anomaly patterns of the sequence numbers
- Reverse labeling method can reconstruct the false routing tree
- Isolating the attackers brings a sharp increase in network performance
- On going research will improve the robustness of the mechanism and the accuracy of identification

Related Ongoing Research

A. Detecting wormhole attacks

B. Position-based private routing in ad hoc networks

C. Time-based private routing in ad hoc networks

D. Congestion aware distance vector (CADV) protocol for ad hoc networks

E. Trust-based Privacy Preservation for Peer-to-peer Data Sharing

E. Trust-based Privacy Preservation for Peer-to-peer Data Sharing

Problem statement

- Privacy in peer-to-peer systems is different from the anonymity problem
- Preserve privacy of requester
- A mechanism is needed to remove the association between the identity of the requester and the data needed

Proposed solution

- A mechanism is proposed that allows the peers to acquire data through trusted proxies to preserve privacy of requester
  - The data request is handled through the peer’s proxies
  - The proxy can become a supplier later and mask the original requester

Related work

- Trust in privacy preservation
  - Authorization based on evidence and trust, [Bhargava and Zhong, DaWaK’02]
  - Developing pervasive trust [Lilien, CGW’03]
- Hiding the subject in a crowd
  - K-anonymity [Sweeney, UFKS’02]
  - Broadcast and multicast [Scarlata *et al*, INCP’01]

Related work (2)

- Fixed servers and proxies
  - Publius [Waldman *et al*, USENIX’00]
- Building a multi-hop path to hide the real source and destination
  - FreeNet [Clarke *et al*, IC’02]
  - Crowds [Reiter and Rubin, ACM TISS’98]
  - Onion routing [Goldschlag *et al*, ACM Commu.’99]

Related work (3)

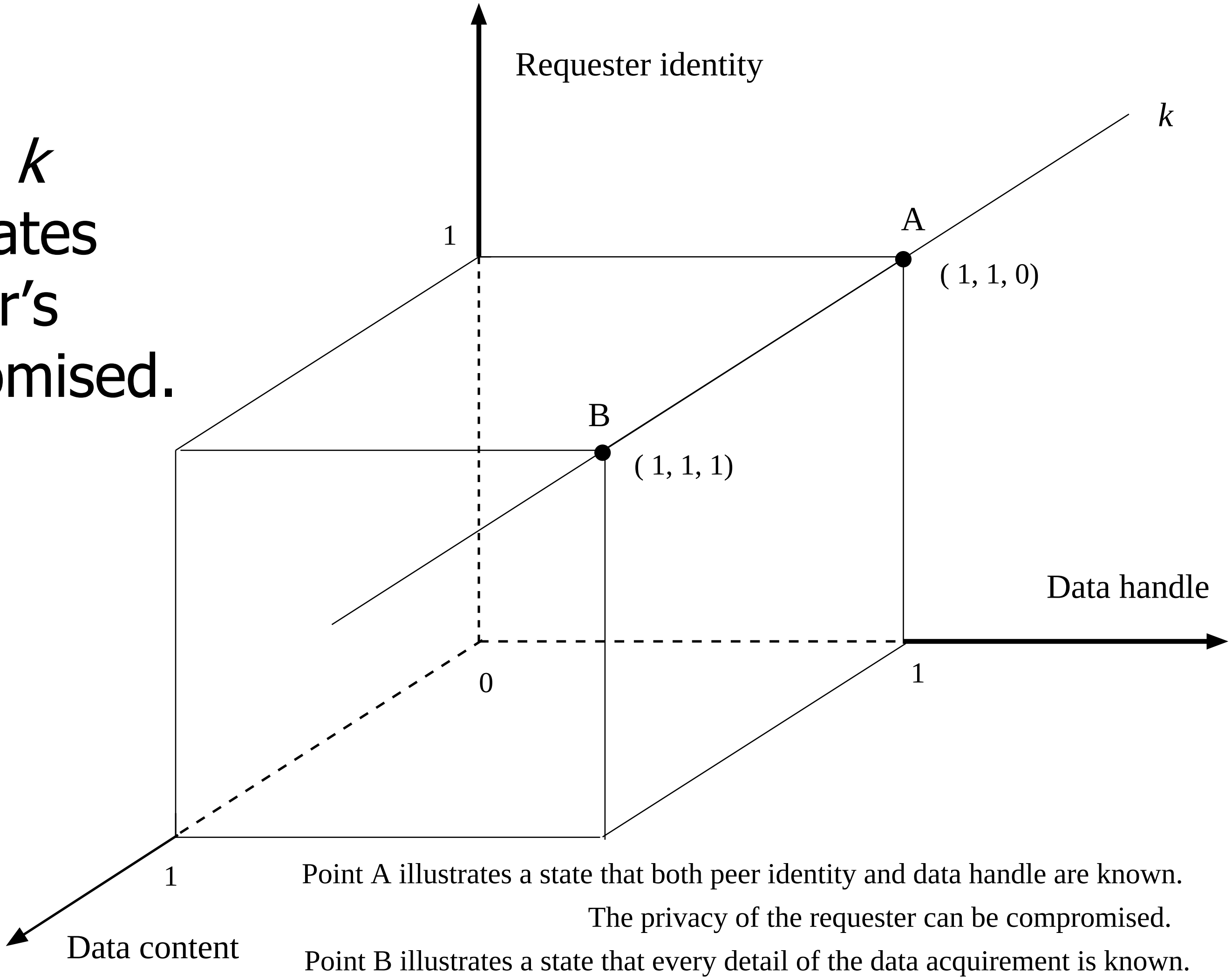
- p^5 [Sherwood *et al*, IEEE SSP'02]
  - p^5 provides sender-receiver anonymity by transmitting packets to a broadcast group
- Herbivore [Goel *et al*, Cornell Univ Tech Report'03]
  - Provides provable anonymity in peer-to-peer communication systems by adopting dining cryptographer networks

Privacy measurement

- A tuple <requester ID, data handle, data content> is defined to describe a data acquirement.
- For each element, “0” means that the peer knows nothing, while “1” means that it knows everything.
- A state in which the requester’s privacy is compromised can be represented as a vector <1, 1, y>, (y Є [0,1]) from which one can link the ID of the requester to the data that it is interested in.

Privacy measurement (2)

For example, line *k* represents the states that the requester's privacy is compromised.



Point A illustrates a state that both peer identity and data handle are known. The privacy of the requester can be compromised.

Point B illustrates a state that every detail of the data acquirement is known.

Mitigating collusion

- An operation “*” is defined as:

$$< c_1, c_2, c_3 > = < a_1, a_2, a_3 > * < b_1, b_2, b_3 >$$

$$c_i = \begin{cases} \max(a_i, b_i), & a_i \neq 0 \;\; and \;\; b_i \neq 0; \\ 0, & otherwise. \end{cases}$$

- This operation describes the revealed information after a collusion of two peers when each peer knows a part of the “secret”.
- The number of collusions required to compromise the secret can be used to evaluate the achieved privacy

Trust based privacy preservation scheme

- The requester asks one proxy to look up the data on its behalf. Once the supplier is located, the proxy will get the data and deliver it to the requester
  - Advantage: other peers, including the supplier, do not know the real requester
  - Disadvantage: The privacy solely depends on the trustworthiness and reliability of the proxy

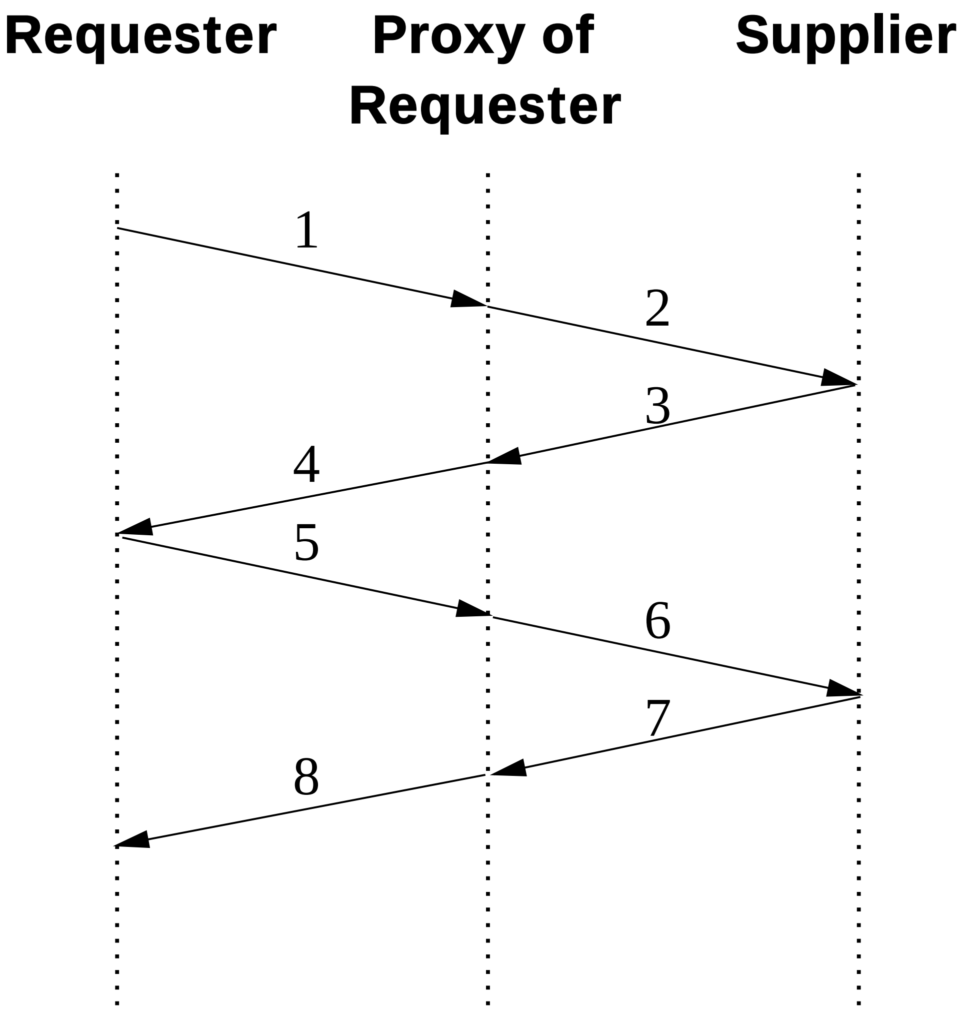
Trust based scheme – Improvement 1

- To avoid specifying the data handle in plain text, the requester calculates the hash code and only reveals a part of it to the proxy.
- The proxy sends it to possible suppliers.
- Receiving the partial hash code, the supplier compares it to the hash codes of the data handles that it holds. Depending on the revealed part, multiple matches may be found.
- The suppliers then construct a bloom filter based on the remaining parts of the matched hash codes and send it back. They also send back their public key certificates.

Trust based scheme – Improvement 1

- Examining the filters, the requester can eliminate some candidate suppliers and finds some who may have the data.
- It then encrypts the full data handle and a data transfer key k_{Data} with the public key.
- The supplier sends the data back using k_{Data} through the proxy
- Advantages:
  - It is difficult to infer the data handle through the partial hash code
  - The proxy alone cannot compromise the privacy
  - Through adjusting the revealed hash code, the allowable error of the bloom filter can be determined

Data transfer procedure after improvement 1



R: requester *S*: supplier

Step 1, 2: *R* sends out the partial hash code of the data handle

Step 3, 4: *S* sends the bloom filter of the handles and the public key certificates

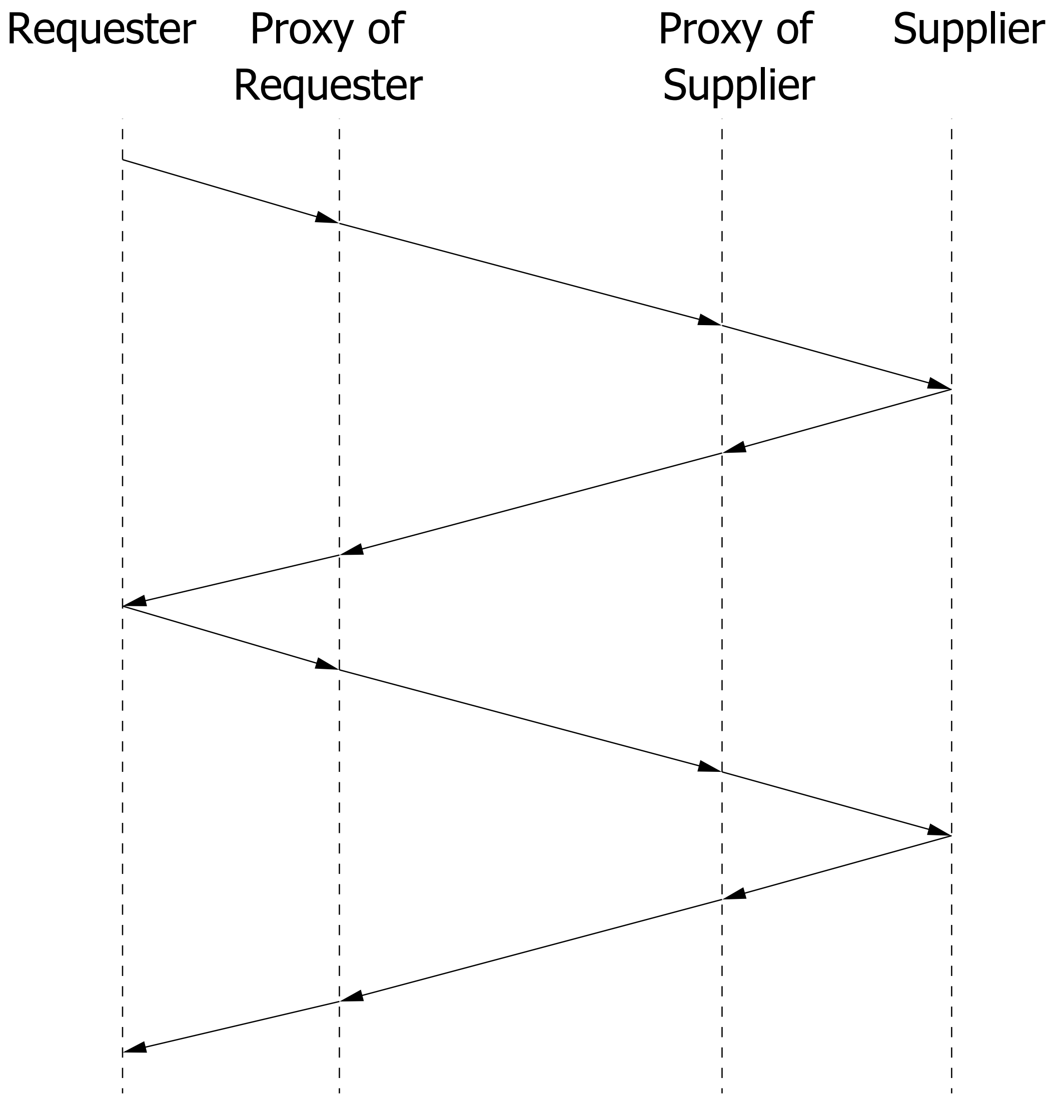
Step 5, 6: *R* sends the data handle and k_{Data} encrypted by the public key

Step 7, 8: *S* sends the required data encrypted by k_{Data}

Trust based scheme – Improvement 2

- The above scheme does not protect the privacy of the supplier
- To address this problem, the supplier can respond to a request via its own proxy

Trust based scheme – Improvement 2



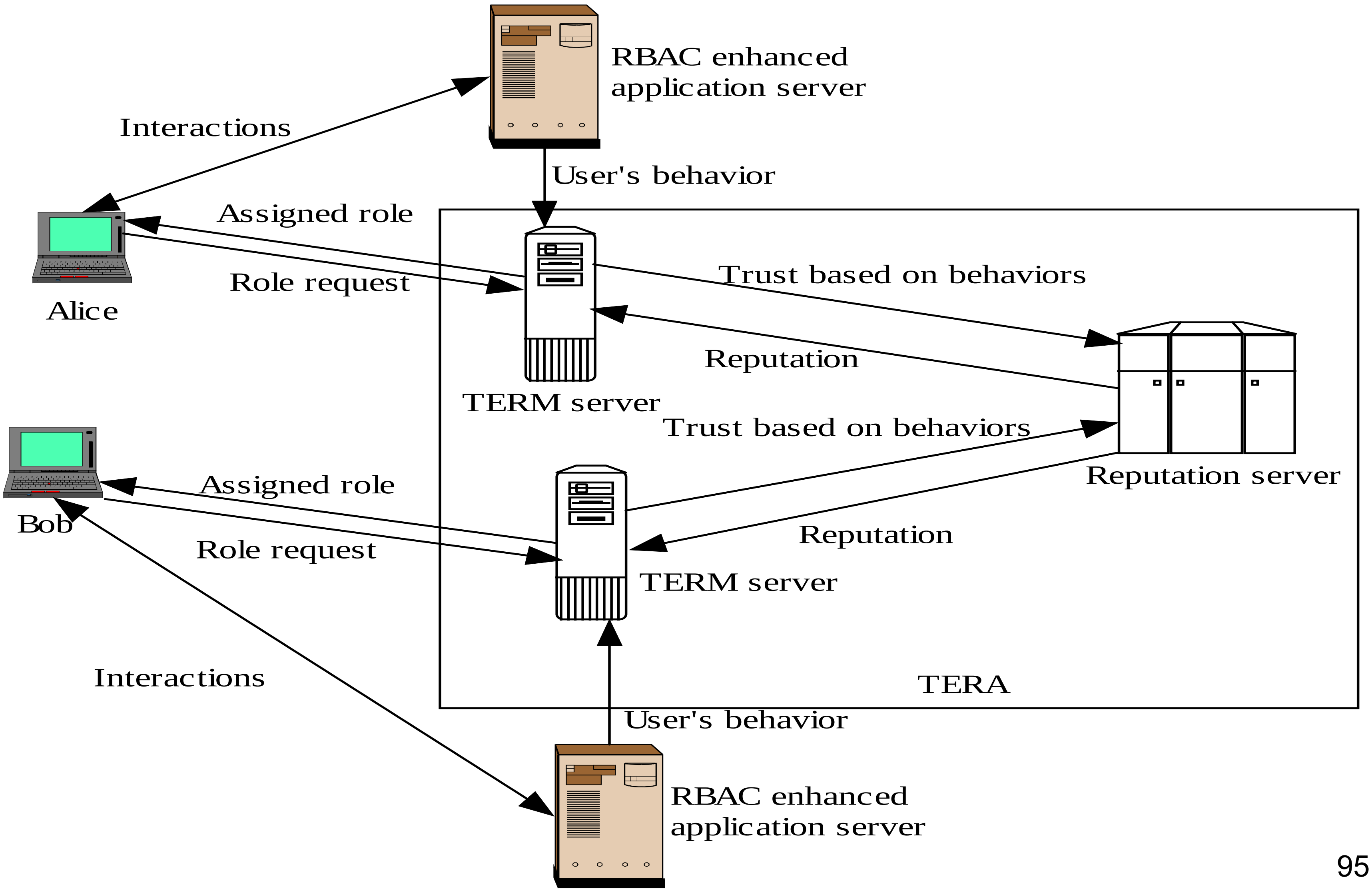
Trustworthiness of peers

- The trust value of a proxy is assessed based on its behaviors and other peers' recommendations
- Using Kalman filtering, the trust model can be built as a multivariate, time-varying state vector

Experimental platform - TERA

- Trust enhanced role mapping (TERM) server assigns roles to users based on
  - Uncertain & subjective evidences
  - Dynamic trust
- Reputation server
  - Dynamic trust information repository
  - Evaluate reputation from trust information by using algorithms specified by TERM server

Trust enhanced role assignment architecture (TERA)



Conclusion

- A trust based privacy preservation method for peer-to-peer data sharing is proposed
- It adopts the proxy scheme during the data acquirement
- Extensions
  - Solid analysis and experiments on large scale networks are required
  - A security analysis of the proposed mechanism is required