An Efficient Anycast Scheme for Discovering K Services in Mobile Ad-hoc Networks

P.J. Lin, C. R. Dow, S. C. Chen, C. J. Li, and S. F. Hwang

Department of Information Engineering and Computer Science, Feng Chia University

10/27/2008

Agenda

- Introduction
- Anycast Scheme
  - Anycast Tree Establishment Scheme
  - Service Information Collection Scheme
  - Service Discovery Scheme
  - Maintenance Schemes
- Experimental Environment and Results
- Conclusions

Introduction (1/2)

- Anycast service
  - Provide a one-to-any bidirectional transmission model
- Mobile ad-hoc networks
  - Multi-hop communications
  - Highly dynamic topology
- Most anycast schemes presented in MANETs are used for discovering one service
- Many distributed applications must work with many servers simultaneously
  - RPC and NTP
  - Threshold cryptography: ITTC project, COCA, and MOCA

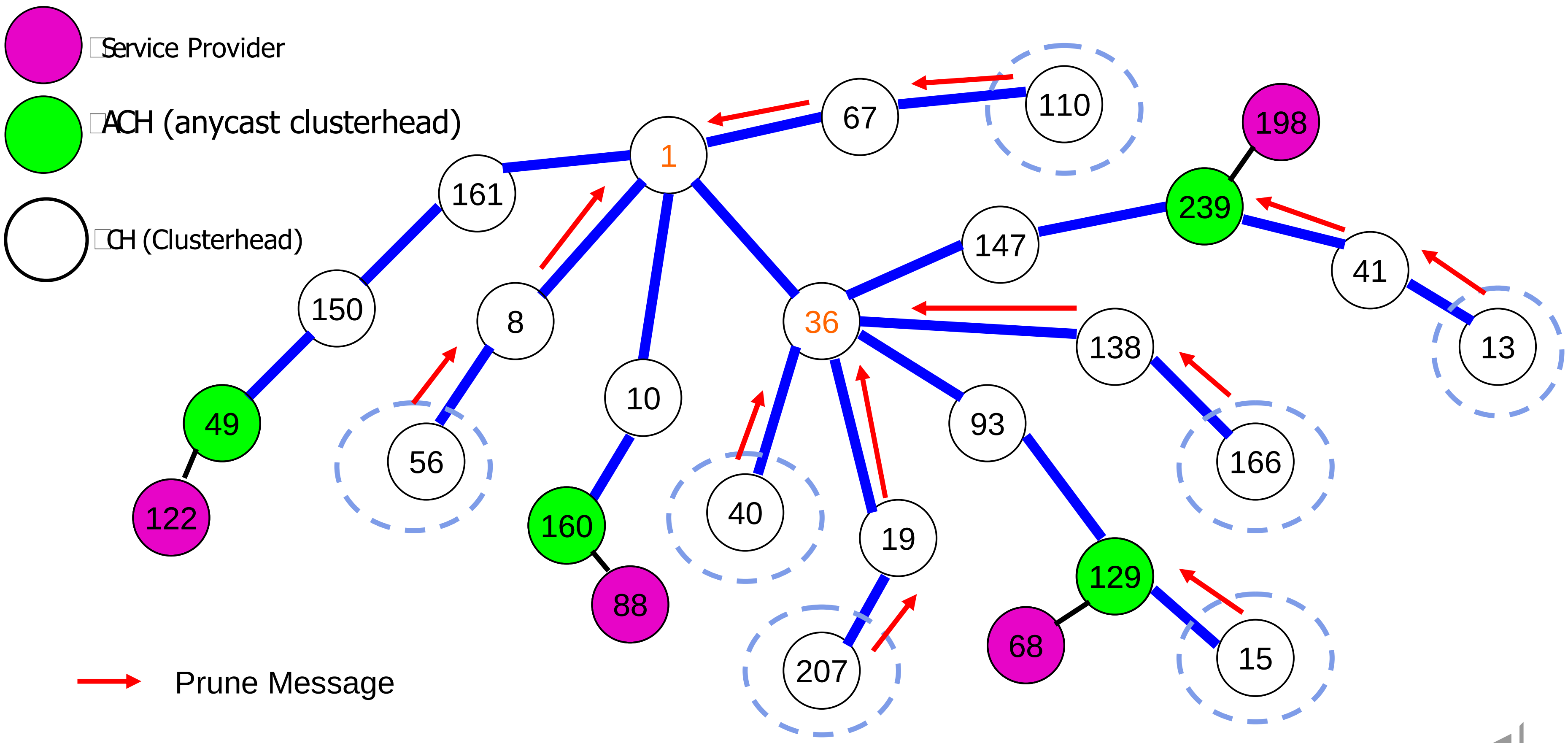
Introduction (2/2)

- There are some extended anycast schemes that can support the discovery of k services
  - Manycast
  - K-anycast
- However, when the senders or receivers increase
  - Low satisfactory ratio, high control overhead, and high searching latency
- Do not have enough service information
  - Locate the service providers
- Lacks a hierarchical structure
  - Reduce the service discovery overhead

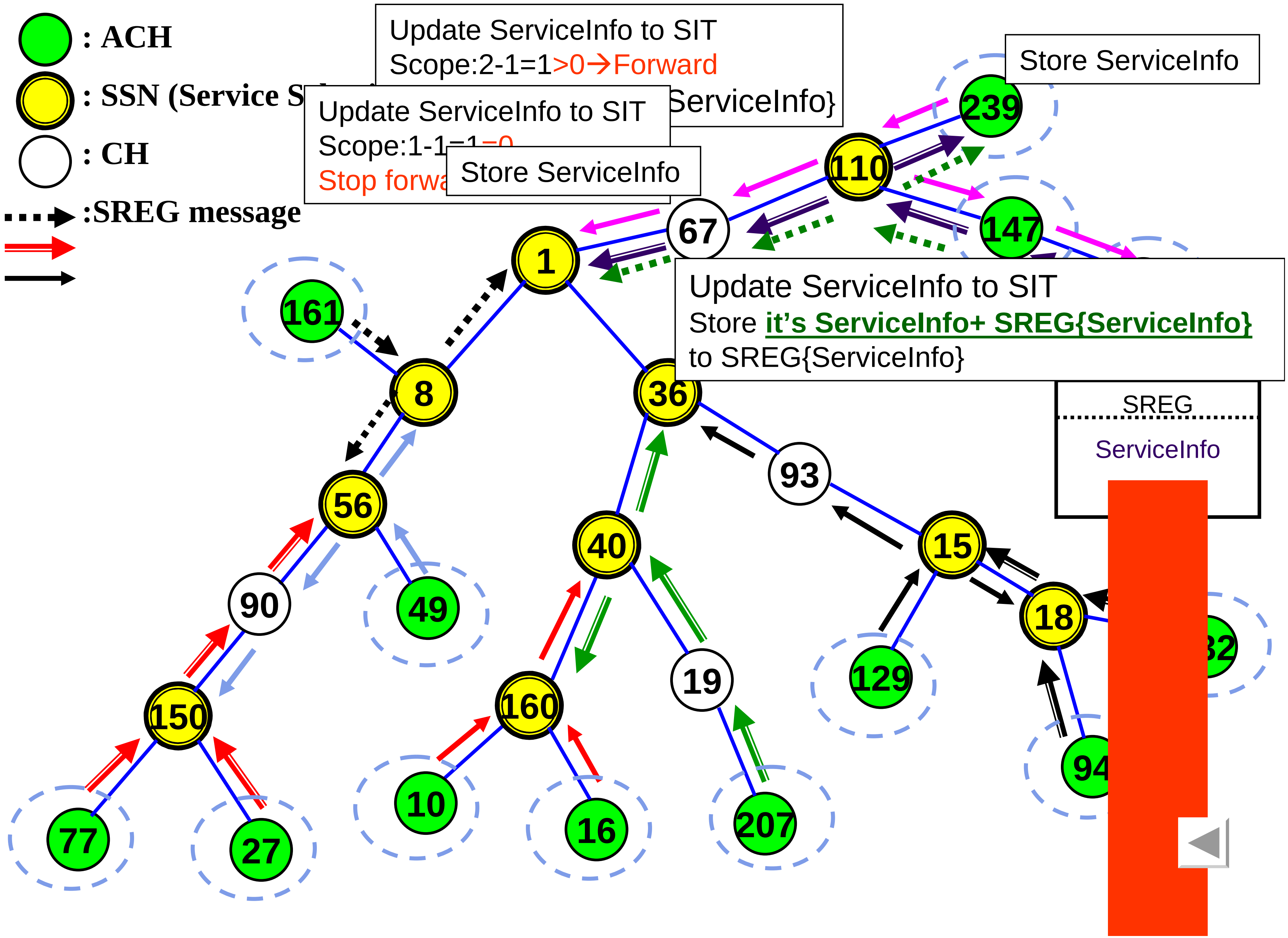
AnyKast Scheme

- In this work, we propose an efficient anycast scheme (AnyKast)
- Anycast tree establishment scheme
  - Avoid unnecessary message transmission
- Service information collection scheme
  - Collect service information for selecting k services
- Service discovery scheme
  - Use service information to search k services
  - Enhance the reliability of our proposed scheme
- Anycast tree and service information maintenance
  - Promote service information accuracy
  - Decrease the control overhead

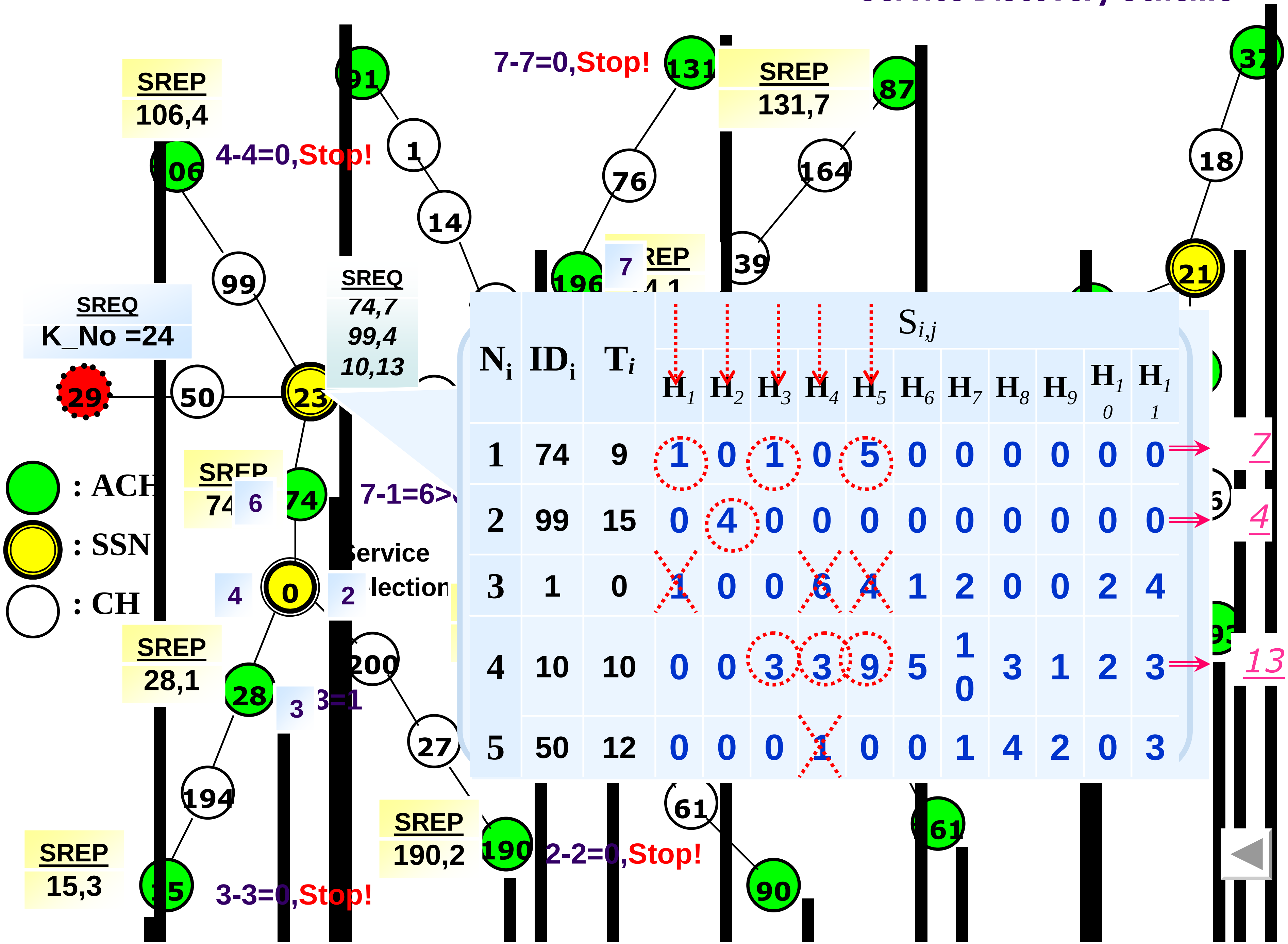
Anycast Tree Establishment Scheme



Service Information Collection Scheme

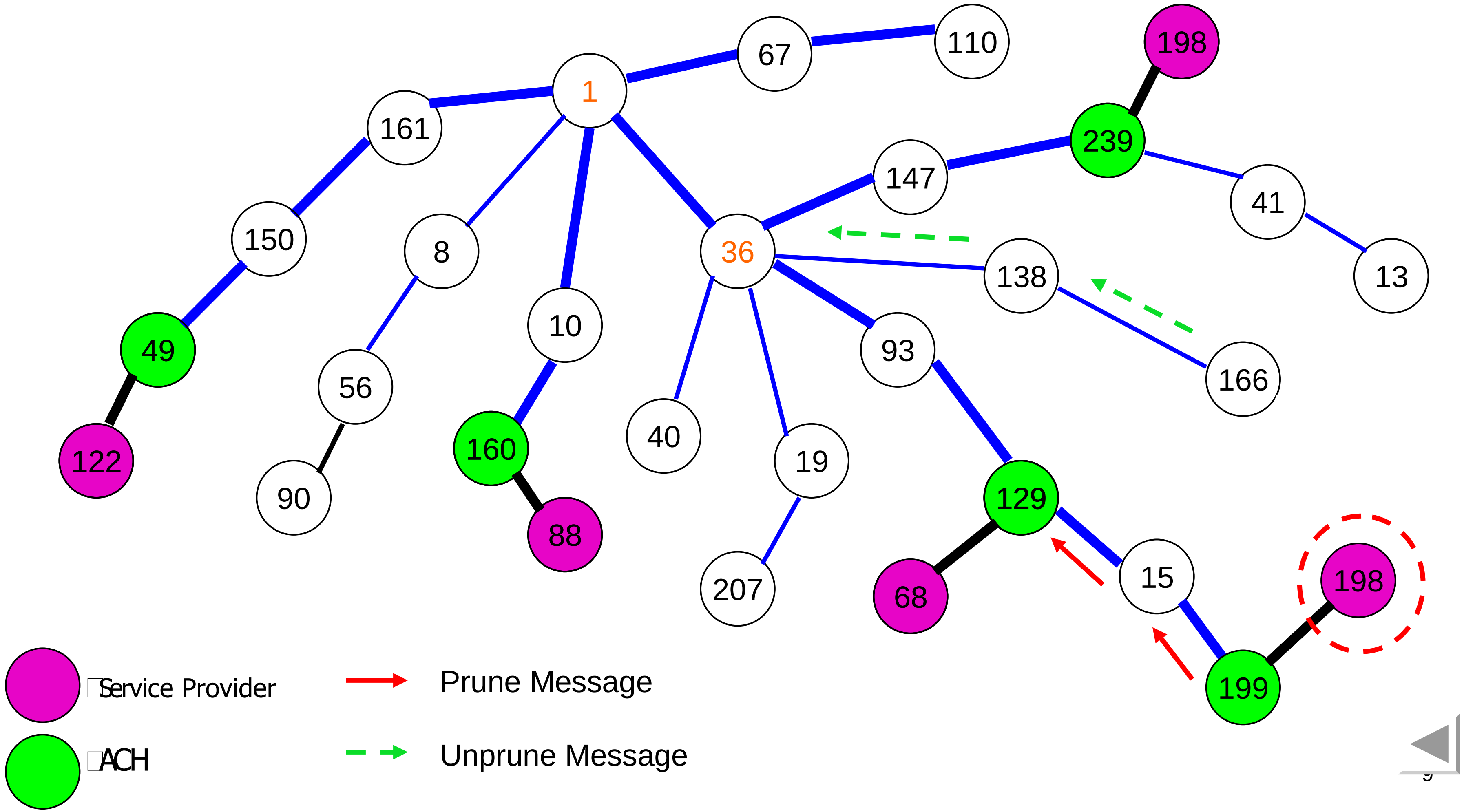


Service Discovery Scheme



N_i	ID_i	T_i	$S_{i,j}$ H_1	H_2	H_3	H_4	H_5	H_6	H_7	H_8	H_9	H_{10}	H_{11}	
1	74	9	1	0	1	0	5	0	0	0	0	0	0	→ 7
2	99	15	0	4	0	0	0	0	0	0	0	0	0	→ 4
3	1	0	1	0	0	6	4	1	2	0	0	2	4	
4	10	10	0	0	3	3	9	5	10	3	1	2	3	→ 13
5	50	12	0	0	0	1	0	0	1	4	2	0	3	

Anycast Tree Maintenance Scheme



Service Information Maintenance

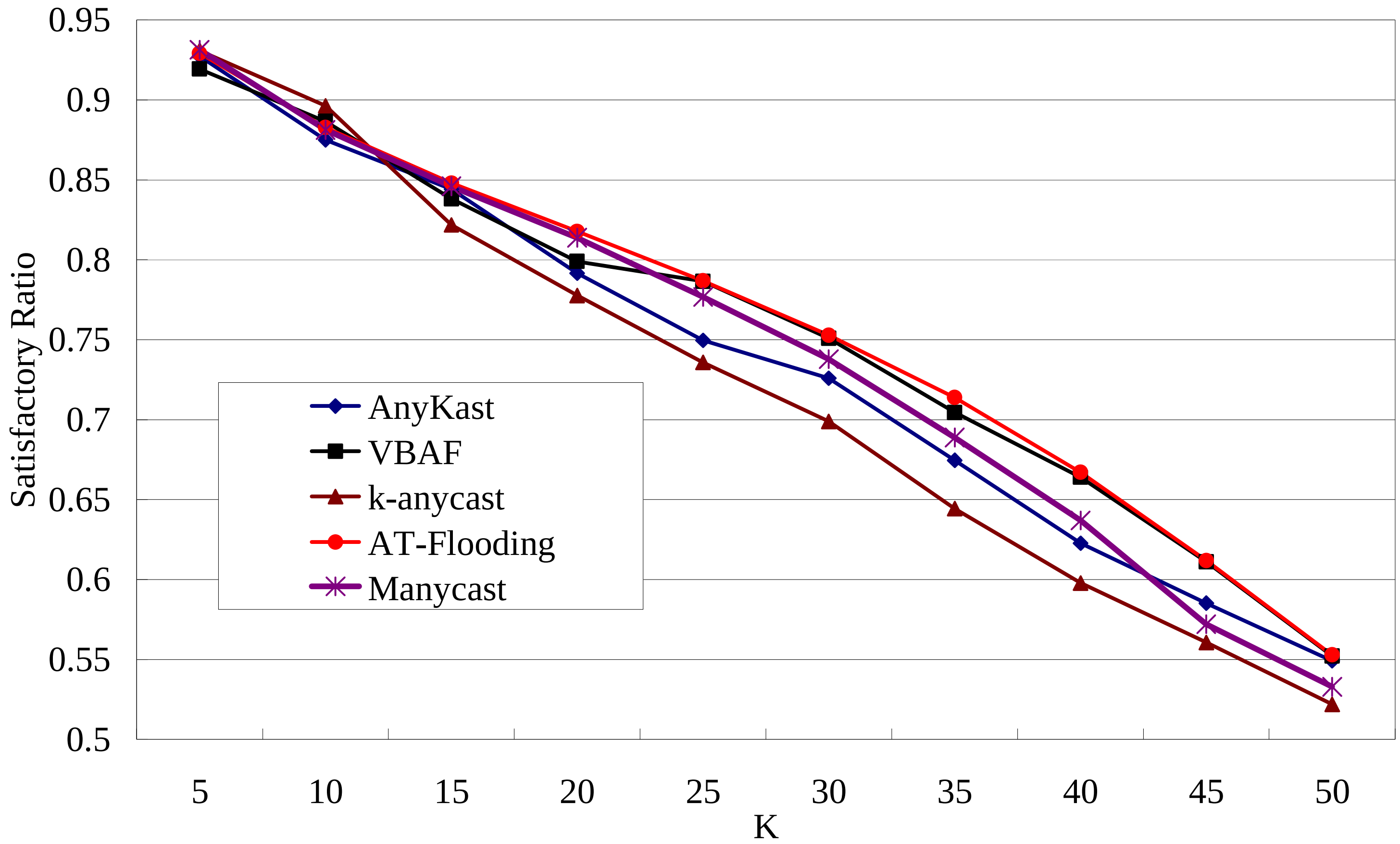
- Service information piggybacking
  - When an ACH or a SSN forwards SREQ and SREP messages, the service information will be piggybacked
- Periodical inquiry
  - When Inquiry Timer is timout
    - A SSN will send INQI message to nodes whose service information is unknown or stale
  - When a SSN or a leaf node receives INQI message
    - Create REPI including its service information and send to sender

Experimental Environment for the AnyKast Scheme

- Simulator: GloMoSim
- Nodes: 250
- Area: 1500m × 1500m
- Tx_range: 250m
- Mobility model: Random waypoint
  - Max Speed: 2m/s
- Number of senders : 1 ~ 10
- Number of services : 25 ~ 125
- Table timer : 20s
- Inquiry timer : 25s

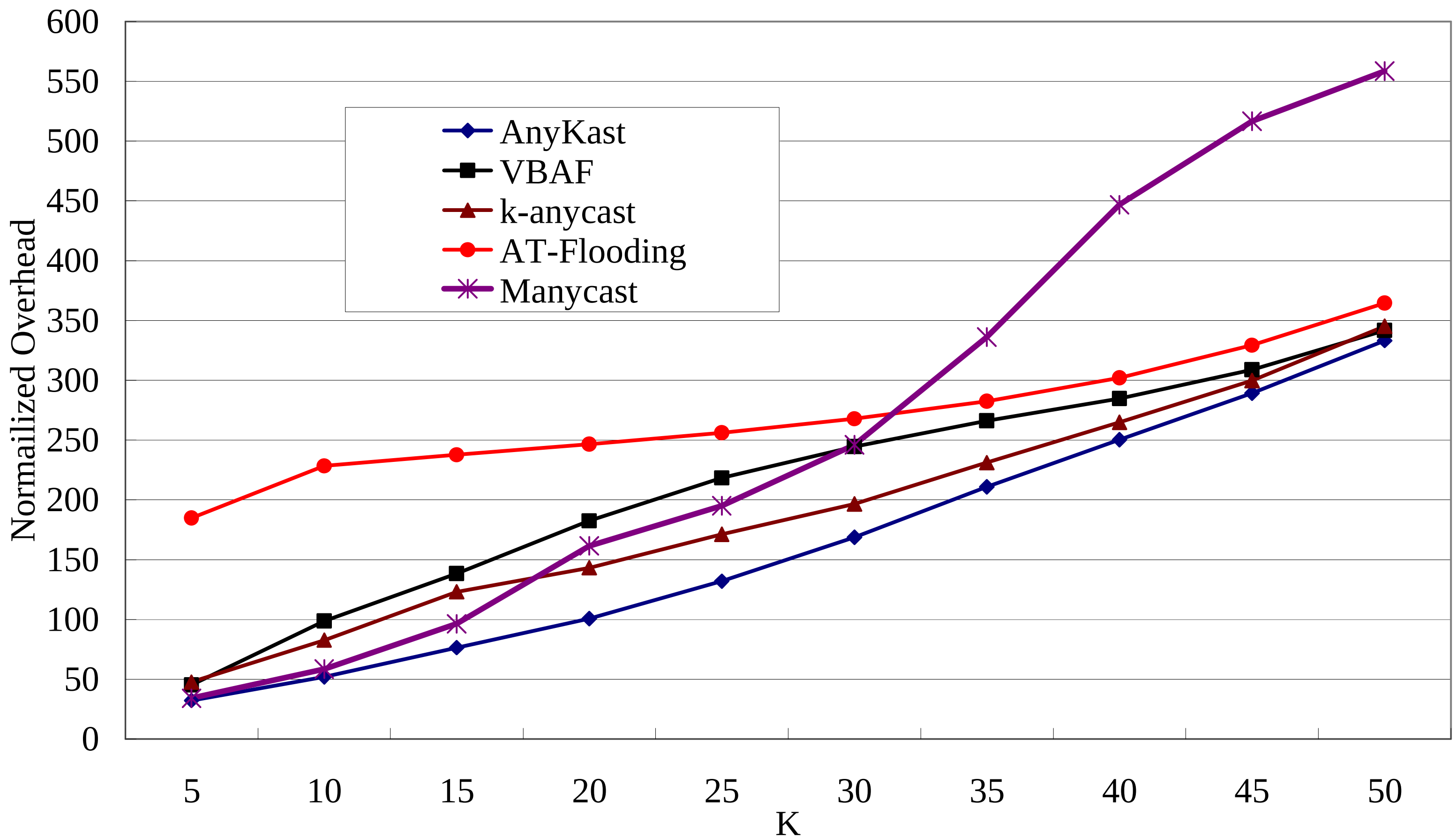
Experimental Results of the AnyKast Scheme (1/4)

- Varying Number of Request Services (1/2)



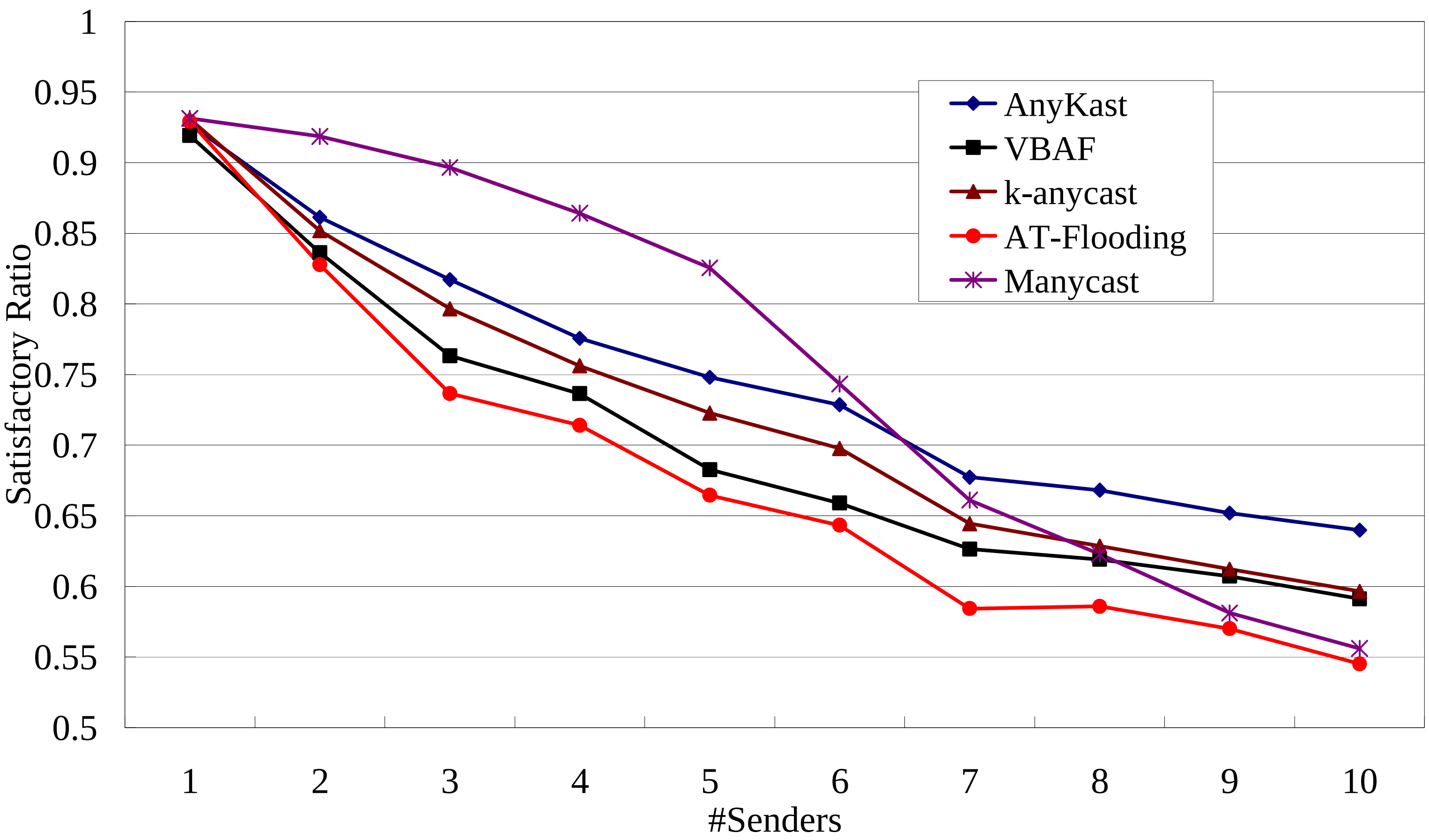
Experimental Results of the AnyKast Scheme (2/4)

- Varying Number of Request Services (2/2)



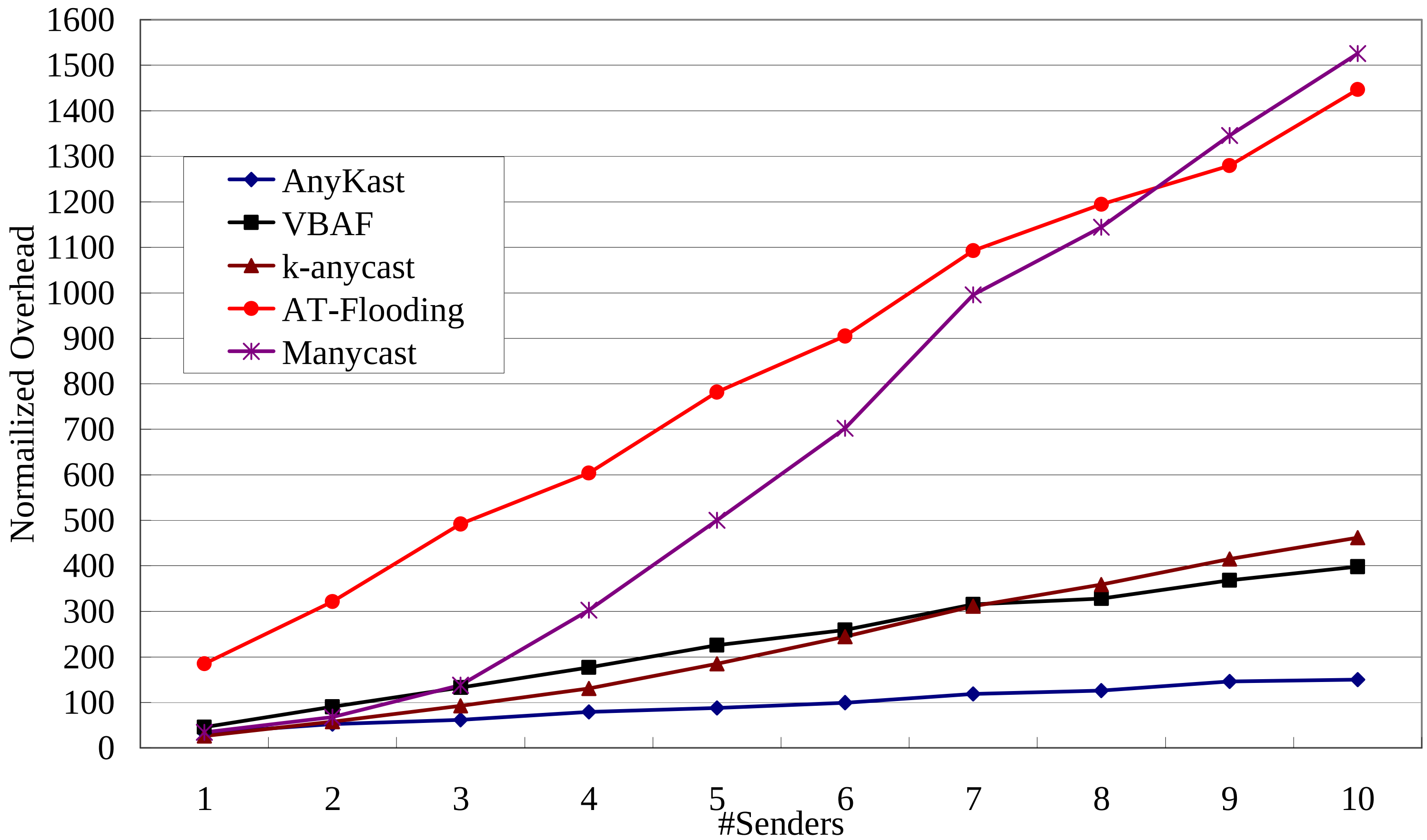
Experimental Results of the AnyKast Scheme (3/4)

- Varying Number of Senders (1/2)



Experimental Results of the AnyKast Scheme (4/4)

- Varying Number of Senders (2/2)



Conclusions

- This work presented an efficient anycast scheme for discovering k services in MANETs
- The established anycast tree can decrease the cost of unnecessary message transmission
- Service discovery scheme use the service information to effectively search k services
- Our maintenance schemes can reduce the control overhead and enhance service information correctness
- The simulation results demonstrate that our AnyKast scheme can effectively discover k services and lower control overhead

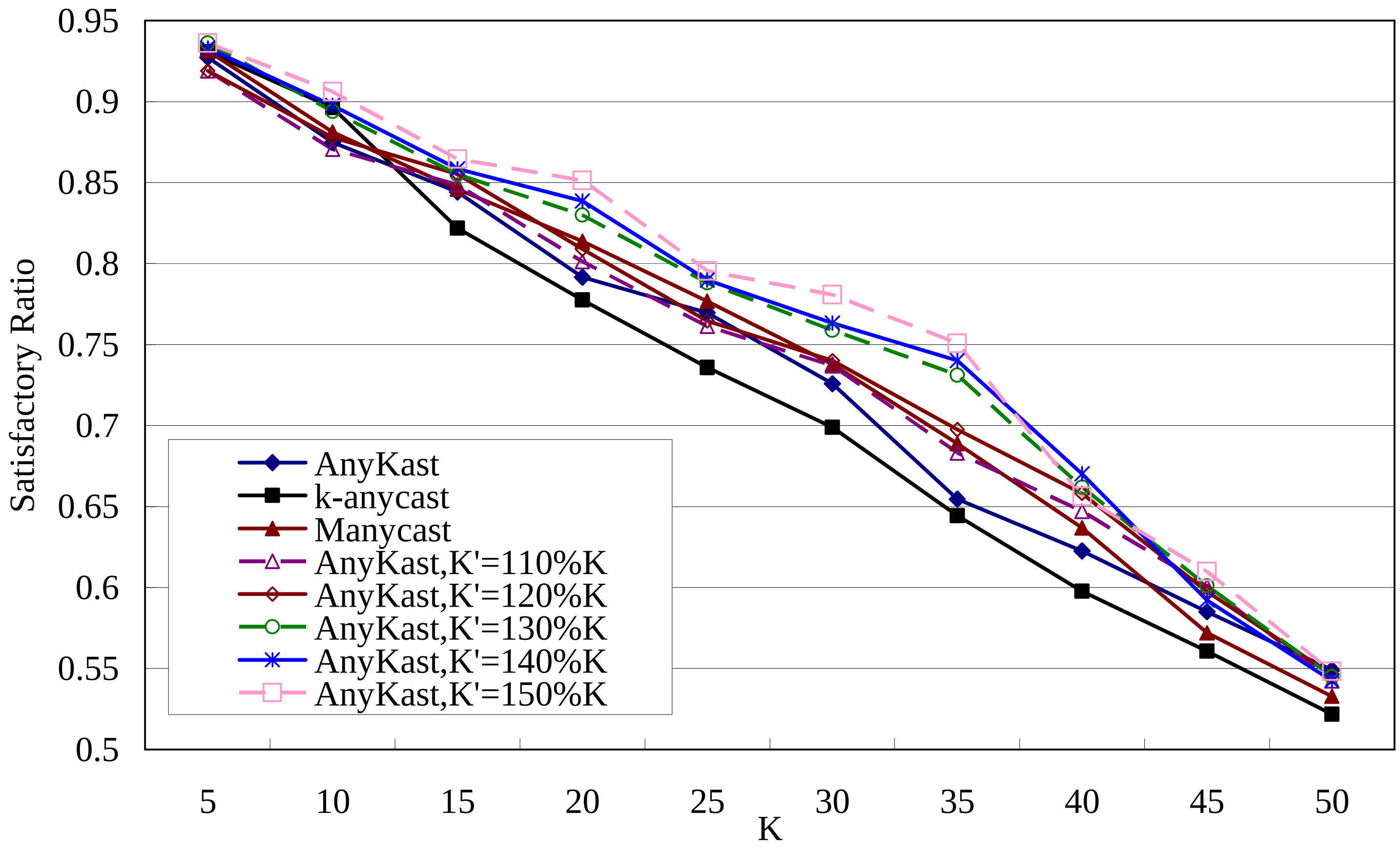
Appendix

Comparisons of Various Schemes

	Hierarchical structure	Collect service information	Discovery strategy
AnyKast	Yes (clustering + virtual backbone)	Yes (SSN)	Scope flooding + Service selection
VBAF	Yes (clustering + virtual backbone)	No	Scope flooding + Forwarding gate
AT-Flooding	Yes (clustering + virtual backbone)	No	Scope flooding
Manycast	No	No	Scope flooding
K-anycast	Yes (simple clustering)	Yes (CH)	Flooding

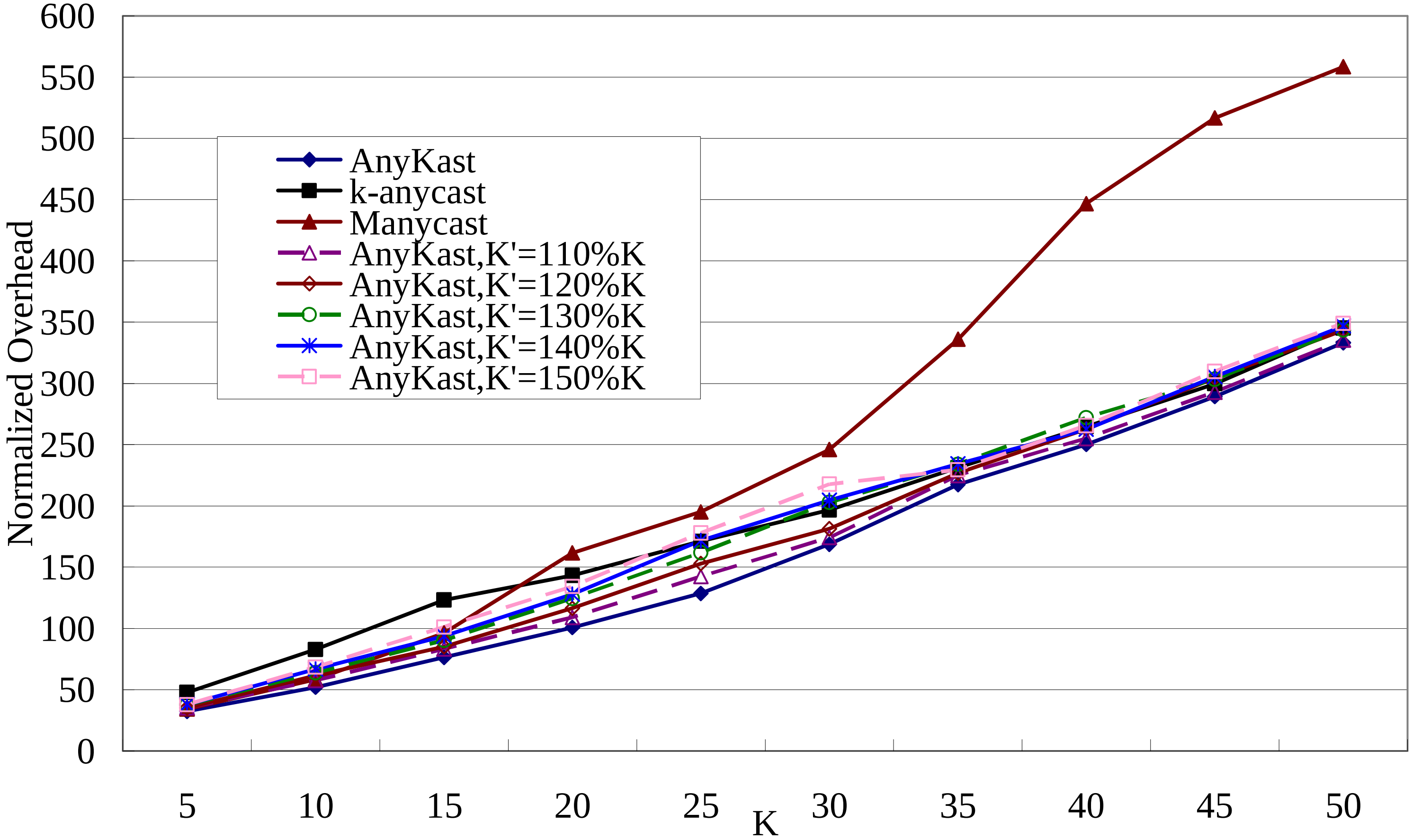
Experimental Results of the AnyKast Scheme (3/6)

- Reliability (1/2)

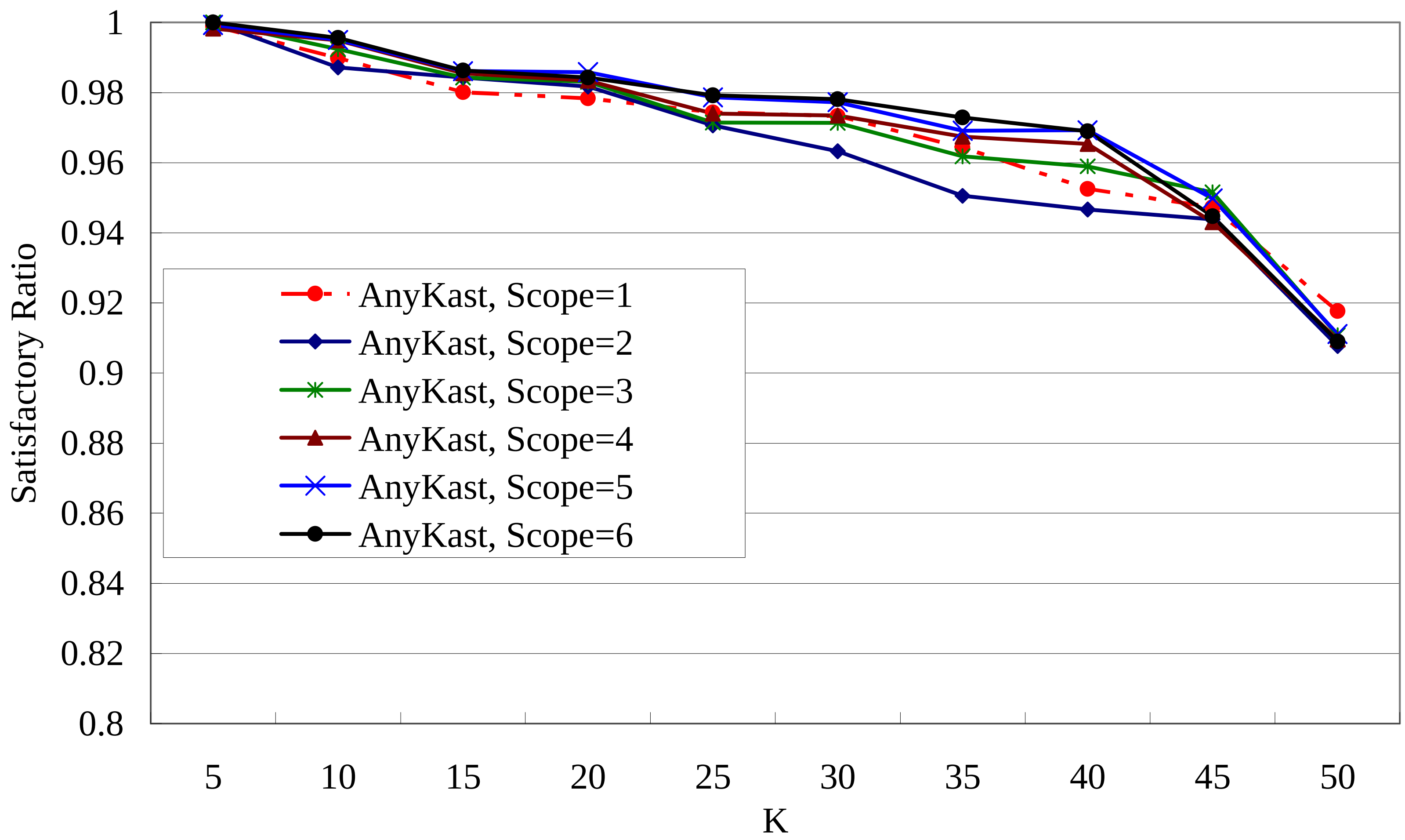


Experimental Results of the AnyKast Scheme (4/6)

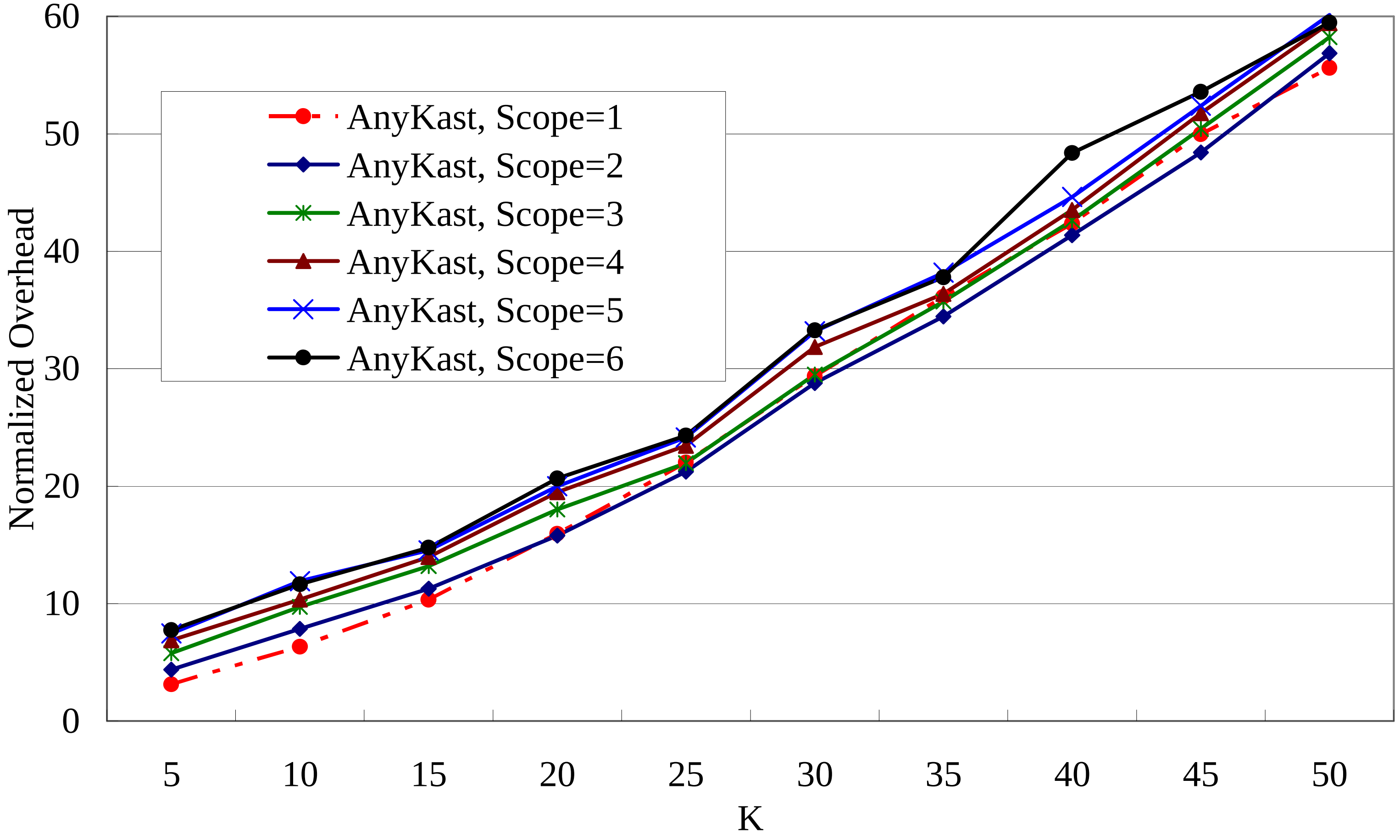
- Reliability (2/2)



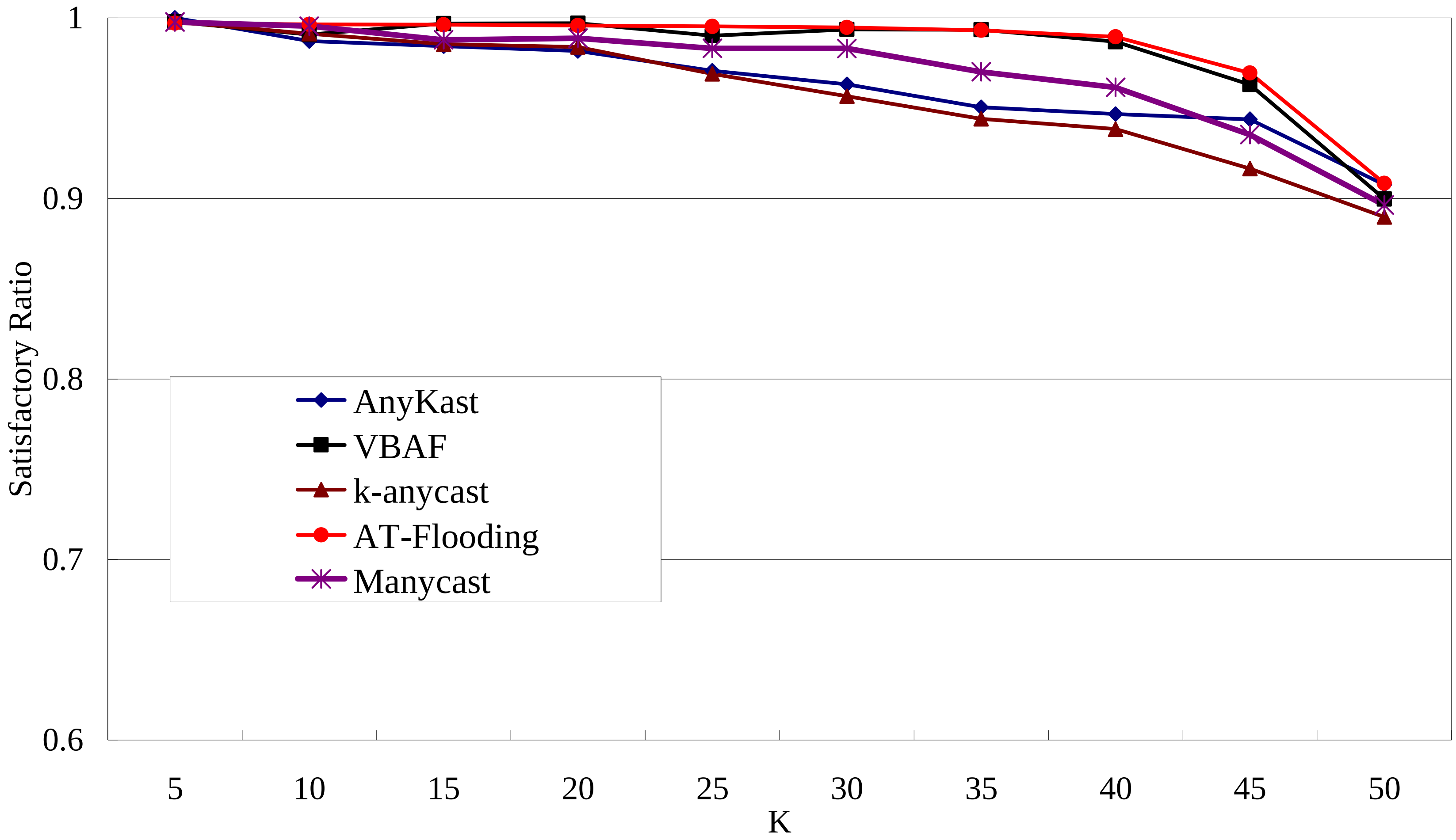
Varying Flooding Scope in Static Ad-hoc Networks



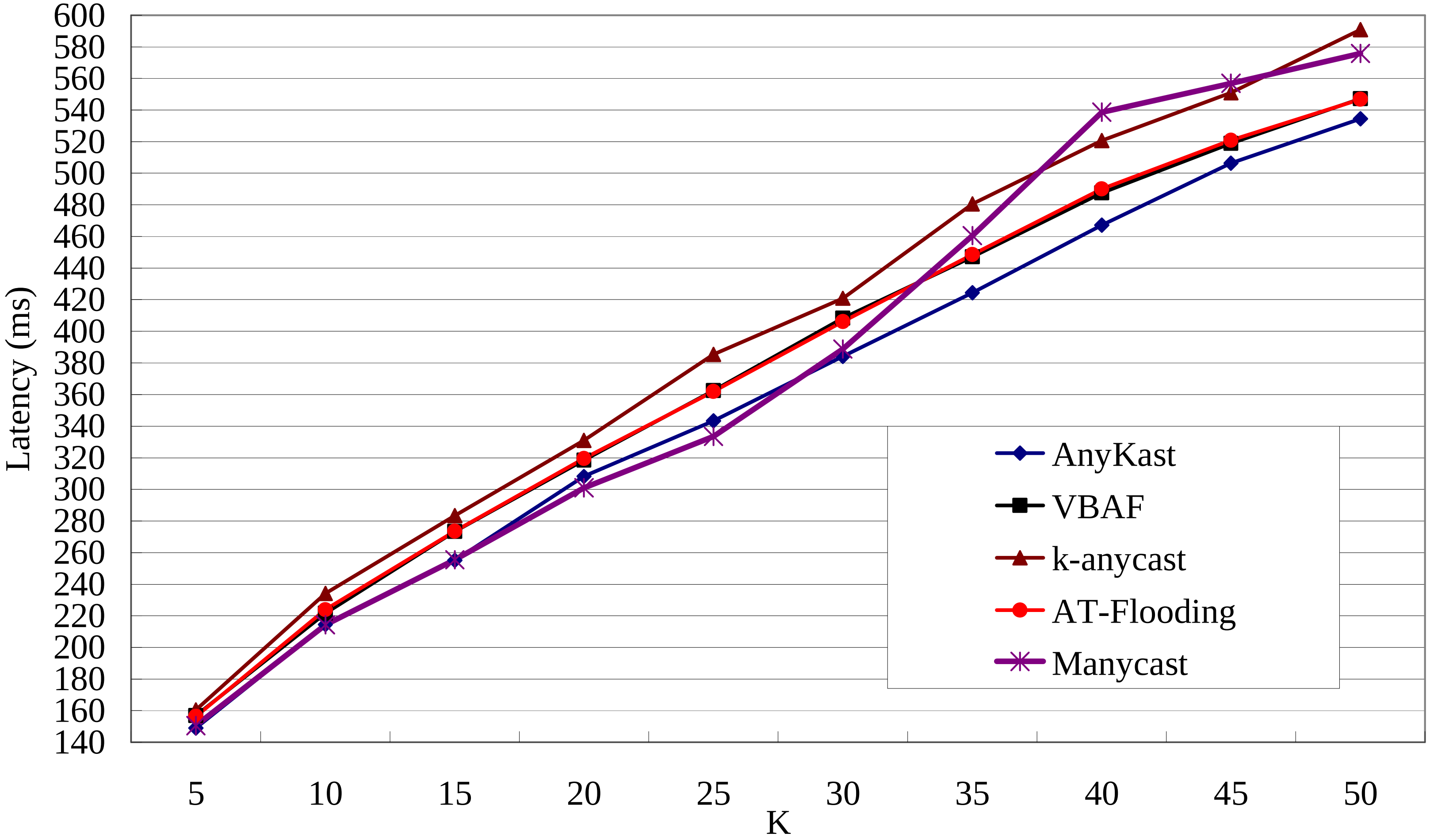
Varying Flooding Scope in Static Ad-hoc Networks



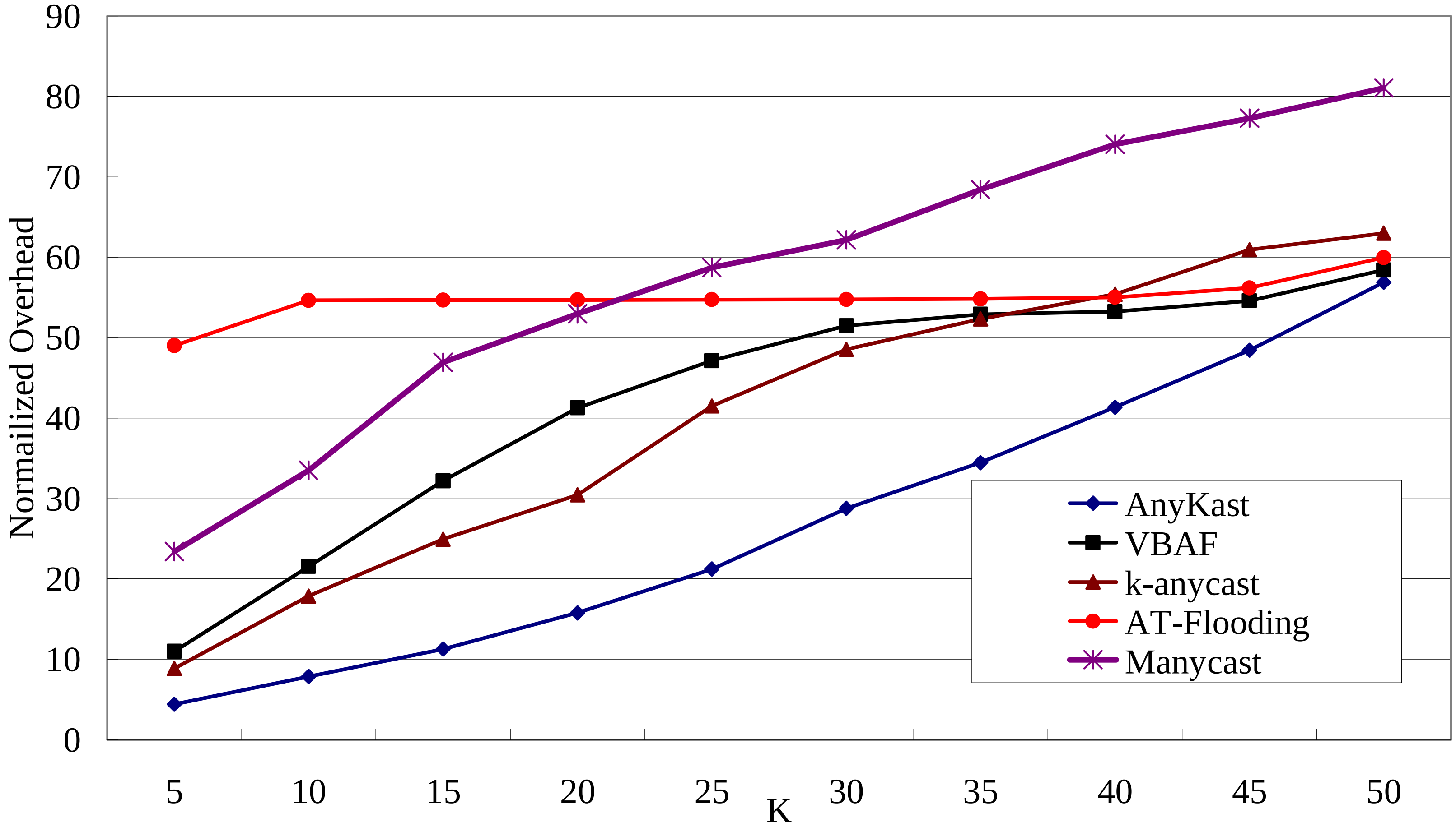
Varying Number of Request Services in Static Ad-hoc Networks



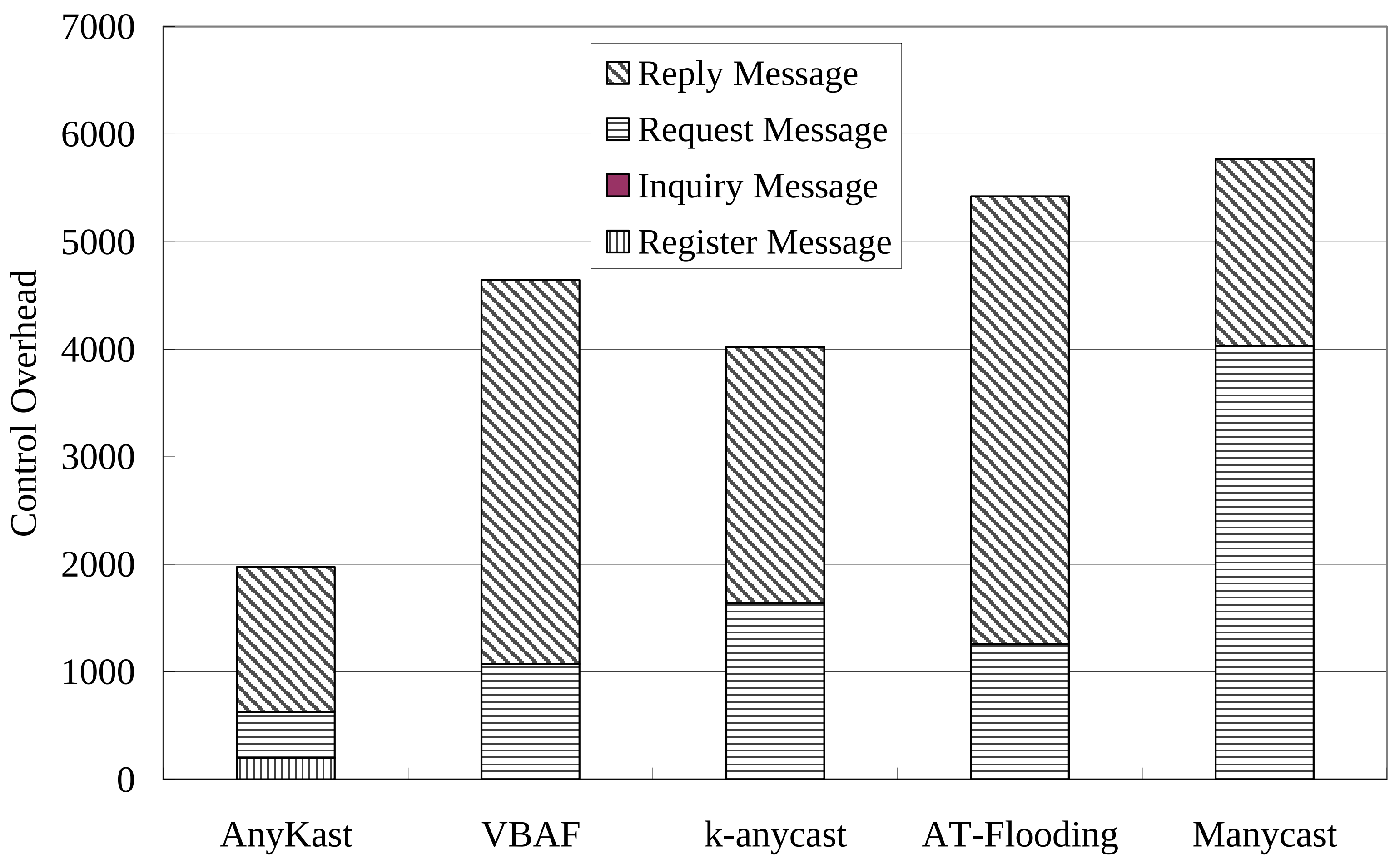
Varying Number of Request Services in Static Ad-hoc Networks



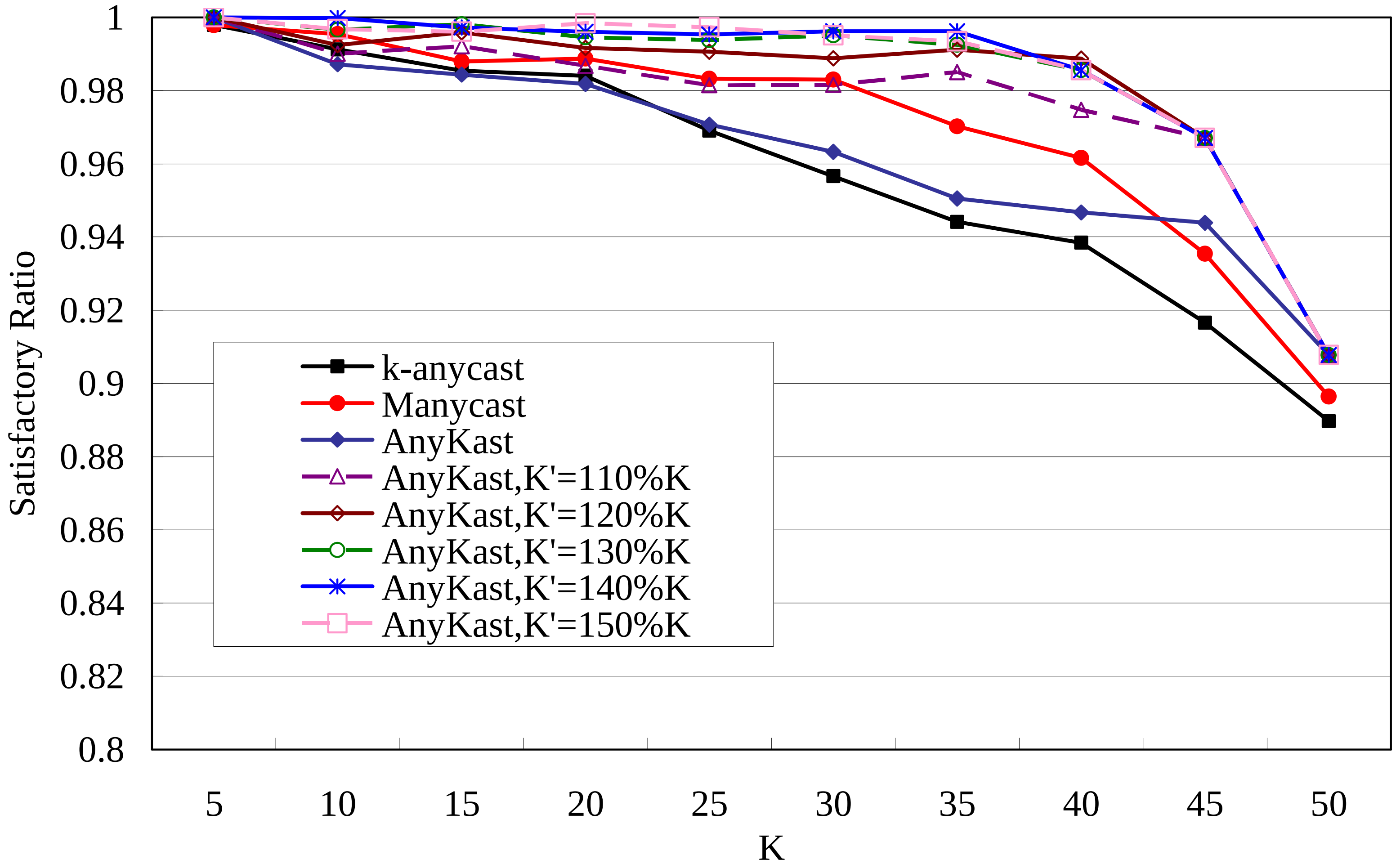
Varying Number of Request Services in Static Ad-hoc Networks



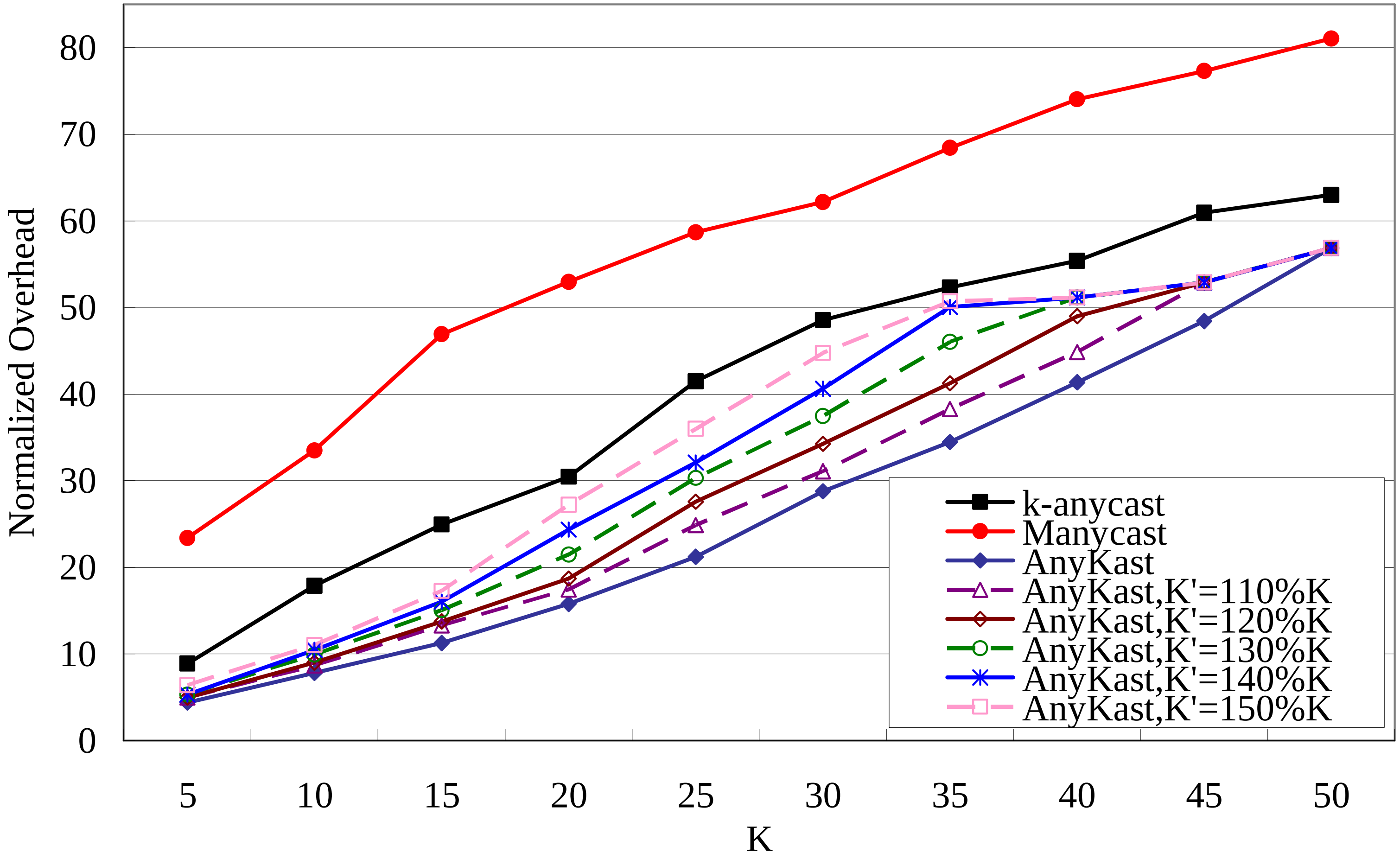
Varying Number of Request Services in Static Ad-hoc Networks



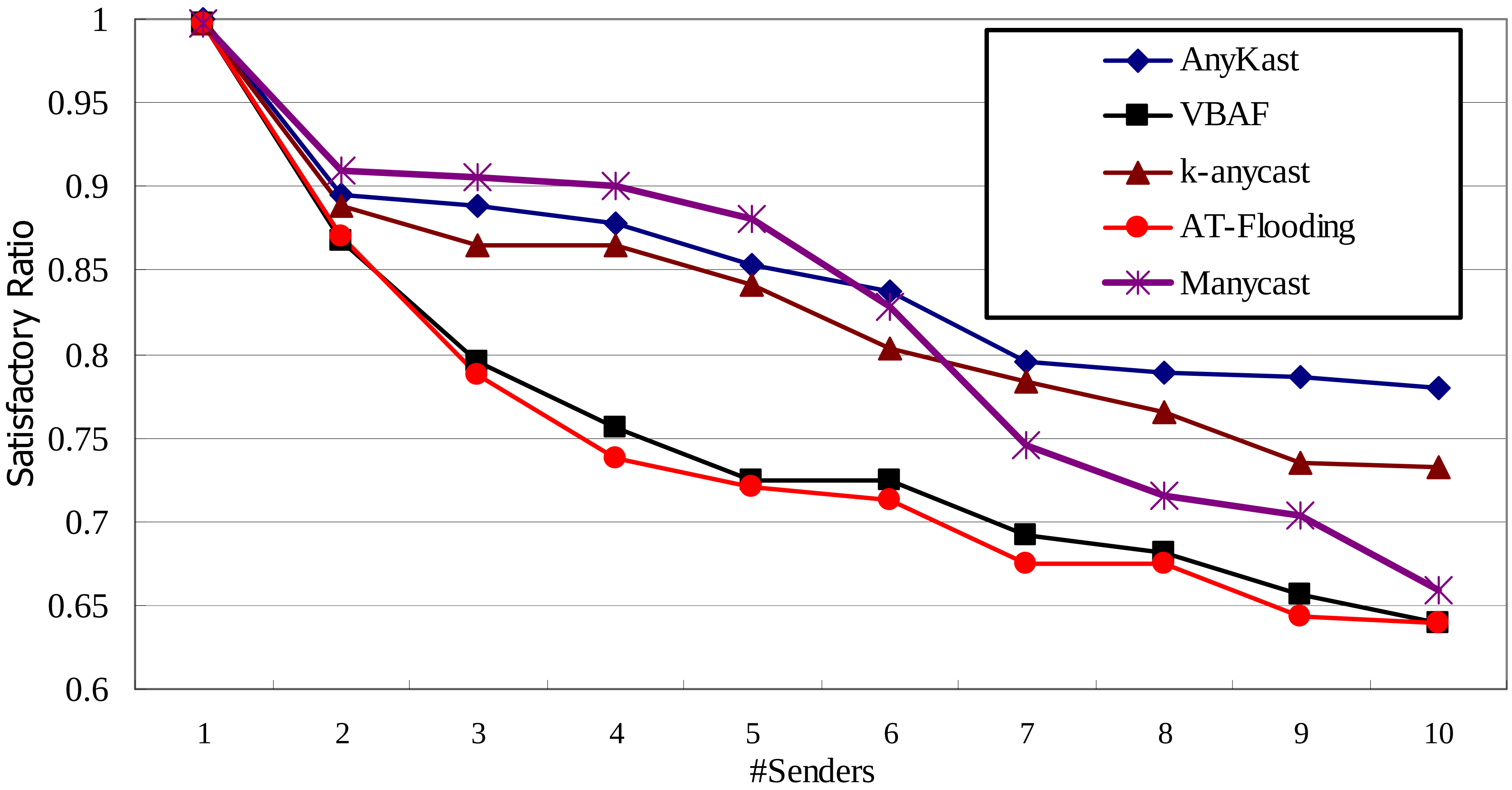
Reliability in Static Ad-hoc Networks



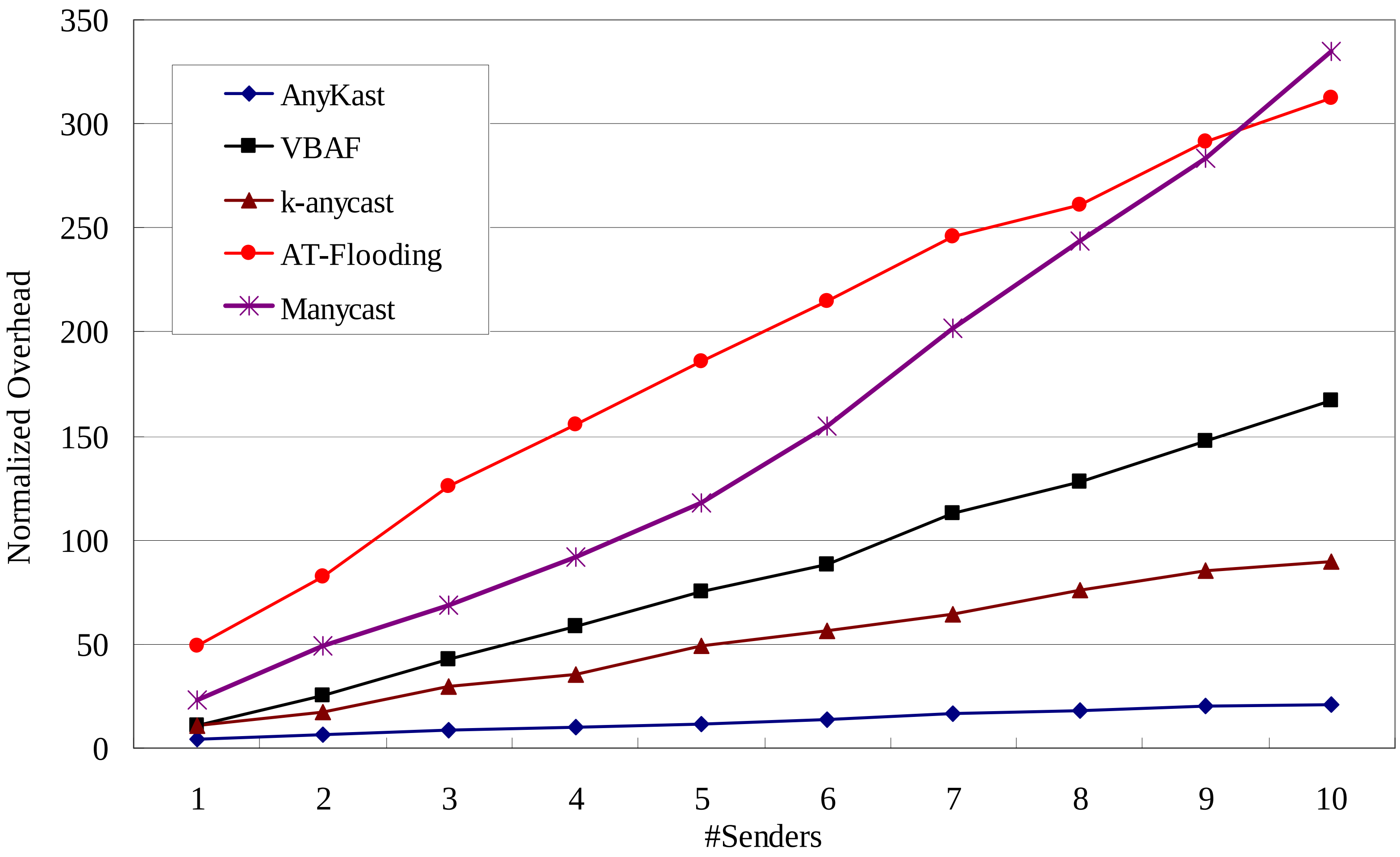
Reliability in Static Ad-hoc Networks



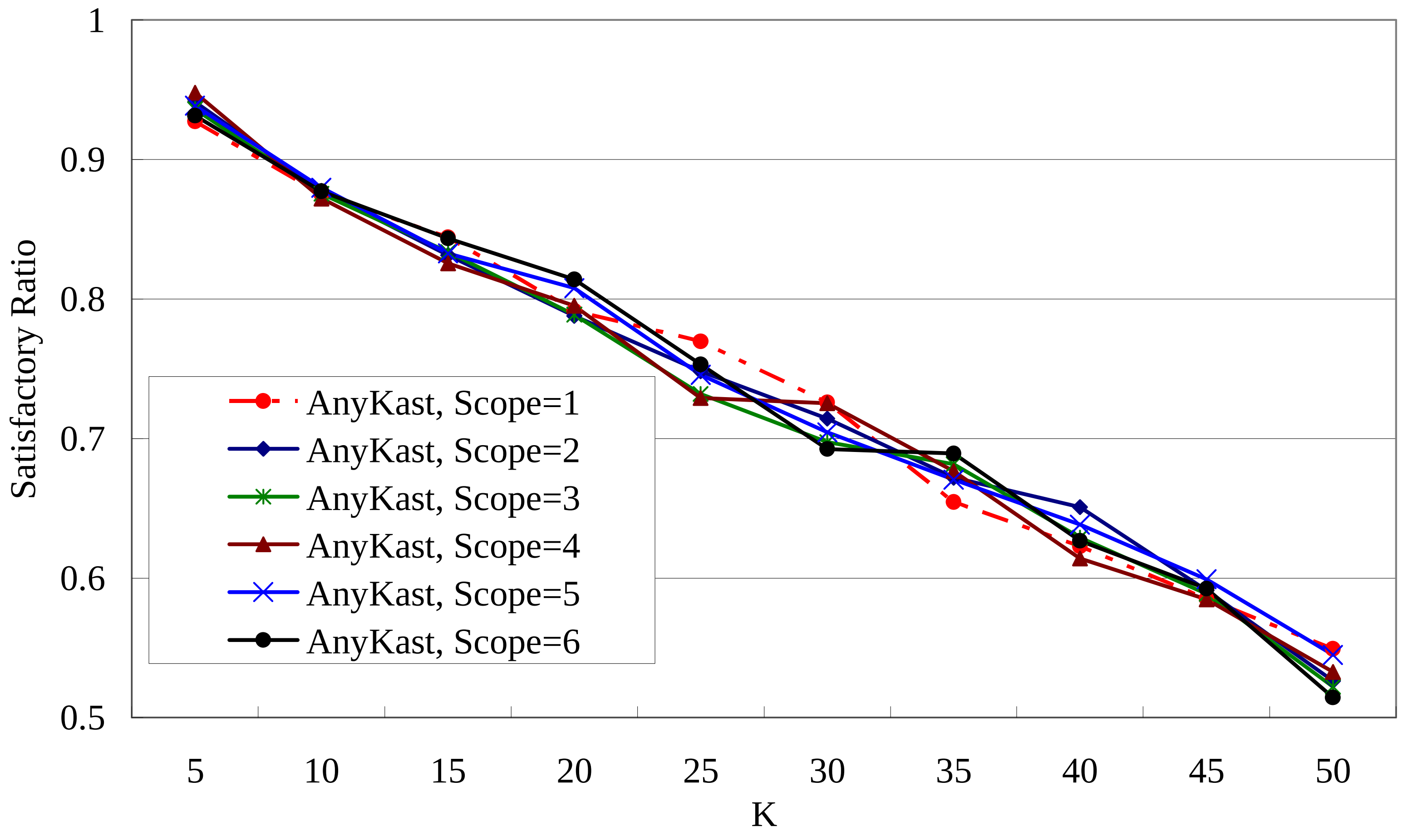
Varying Number of Senders in Static Ad-hoc Networks



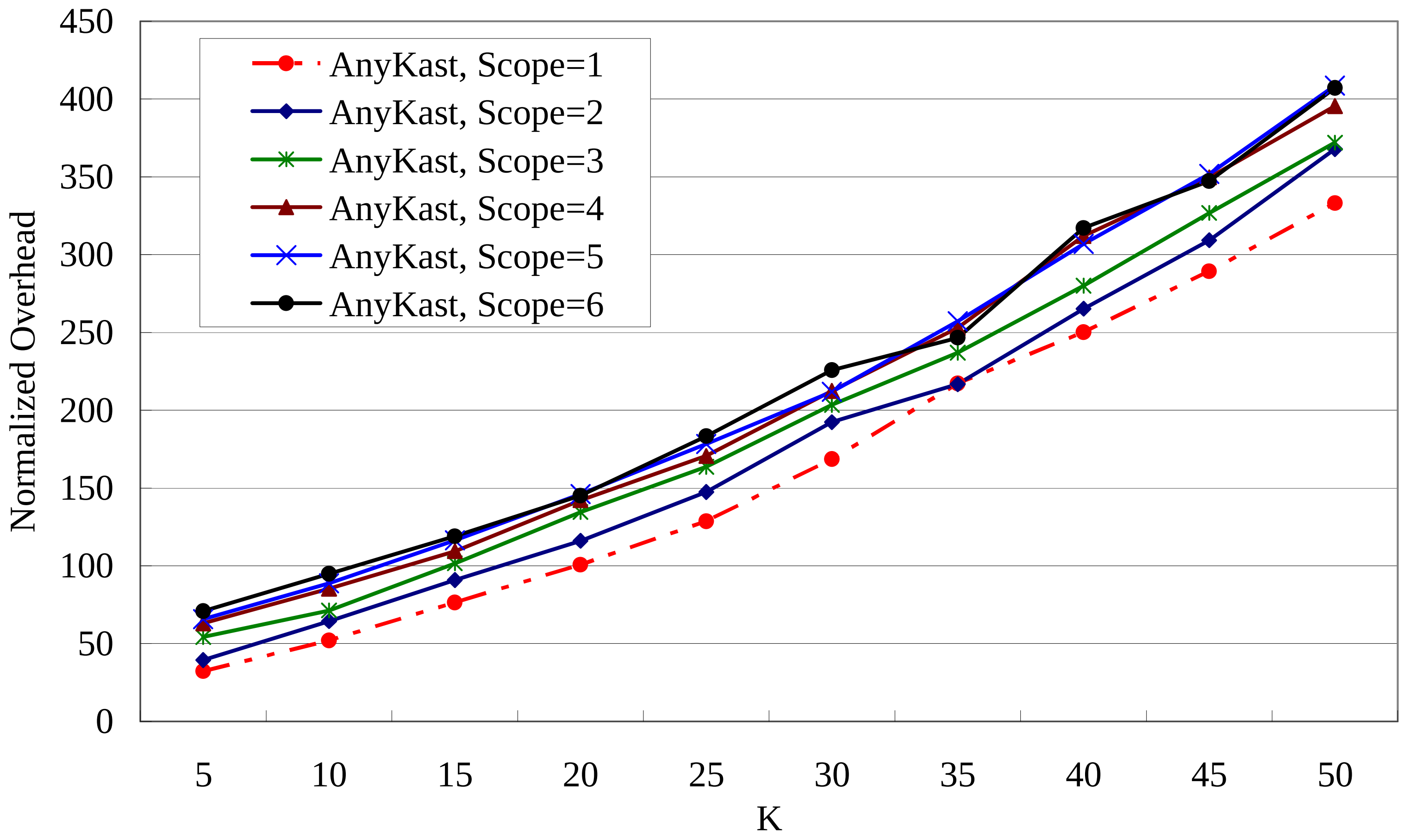
Varying Number of Senders in Static Ad-hoc Networks



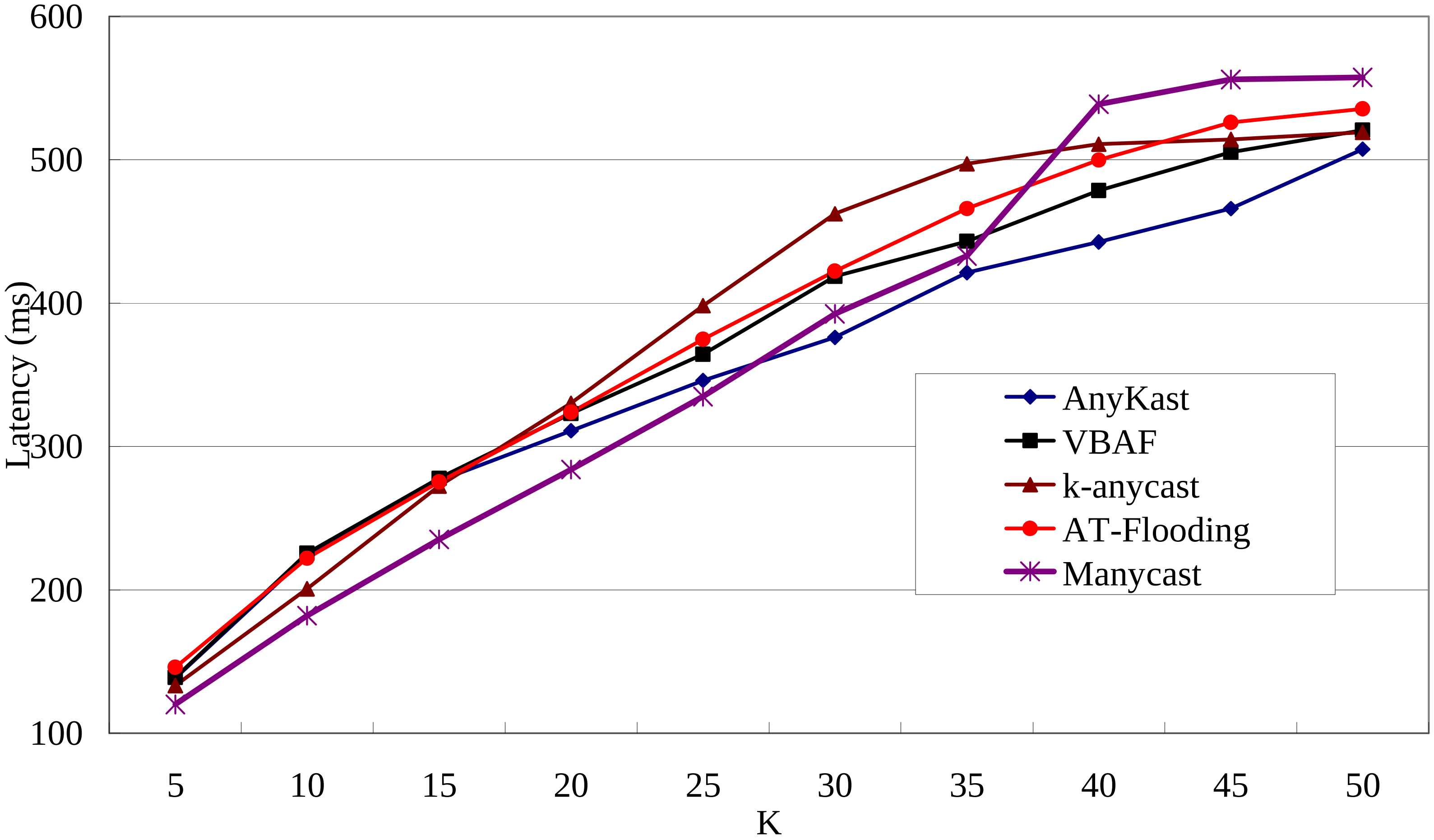
Varying Flooding Scope in Mobile Ad-hoc Networks



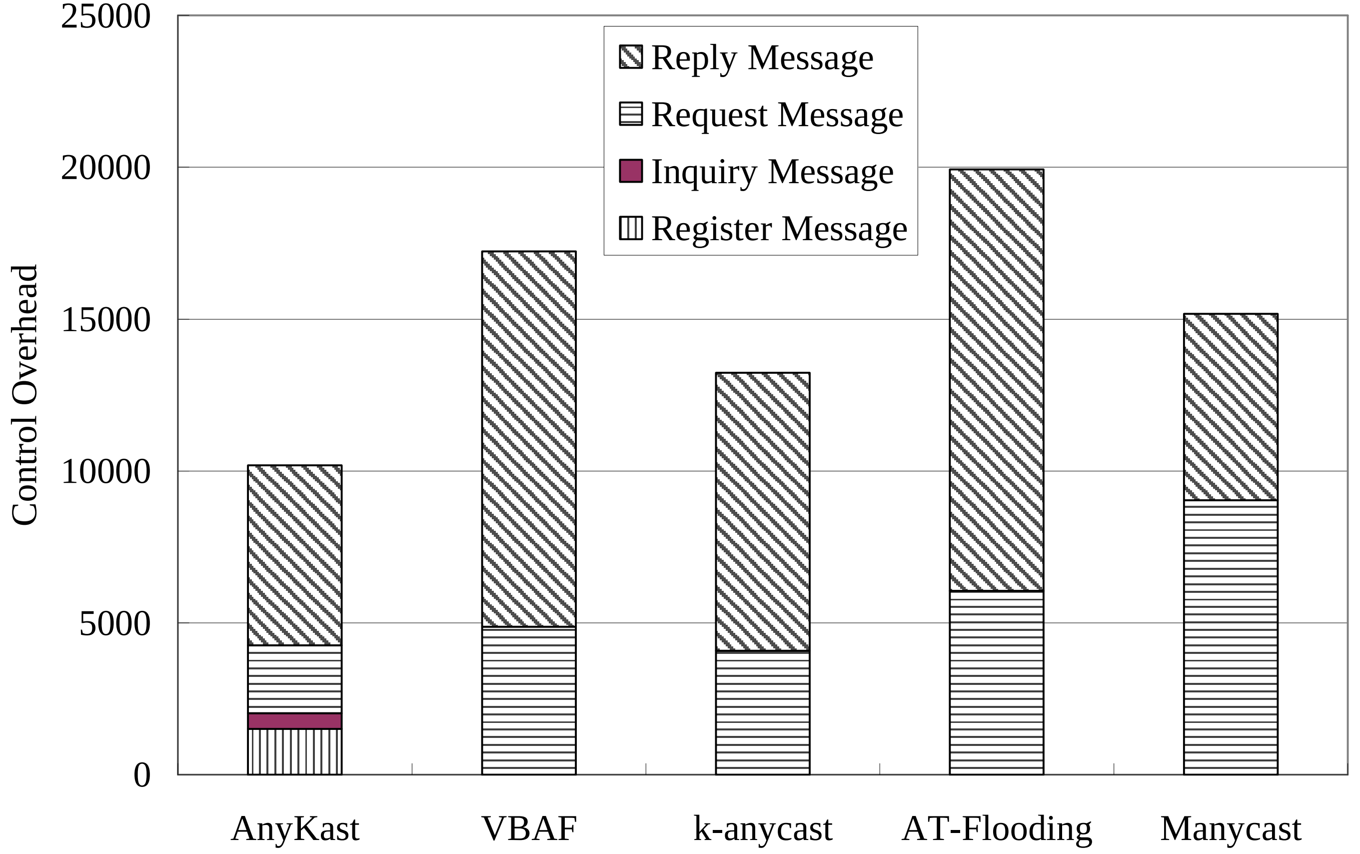
Varying Flooding Scope in Mobile Ad-hoc Networks



Varying Number of Request Services in Mobile Ad-hoc Networks



Varying Number of Request Services in Mobile Ad-hoc Networks



Anycast Tree Establishment Scheme (1/2)

- When the virtual backbone is established in MANETs, the leaf nodes on the virtual backbone can be identified
- When CH $\notin$ **ACH** and Link(CH) = 1
  - CH will send a *PRUNE* message to its parent node
  - Set its *BE_PRUNE* flag to TRUE, and record its parent ID.

Anycast Tree Establishment Scheme (2/2)

- When CH receives the *PRUNE* message
  - it will record that the state of the outgoing link is pruned
  - if CH $\notin$ **ACH** and Link(CH) = 1 (except the pruned link)
    - it will forward a *PRUNE* message to its parent node

Service Information Collection Scheme

- Collect service information as the selection metrics
- When a node becomes an ACH
  - Create a SREG to disseminate its information actively
- Scope flooding
  - The definition of scope
    - The number of SSNs
  - Scope↑
    - Accuracy of service information↑
    - Control overhead ↑

Service Information Collection Scheme (1/2)

- Service Information Table (SIT)
  - Space complexity: O(nm)

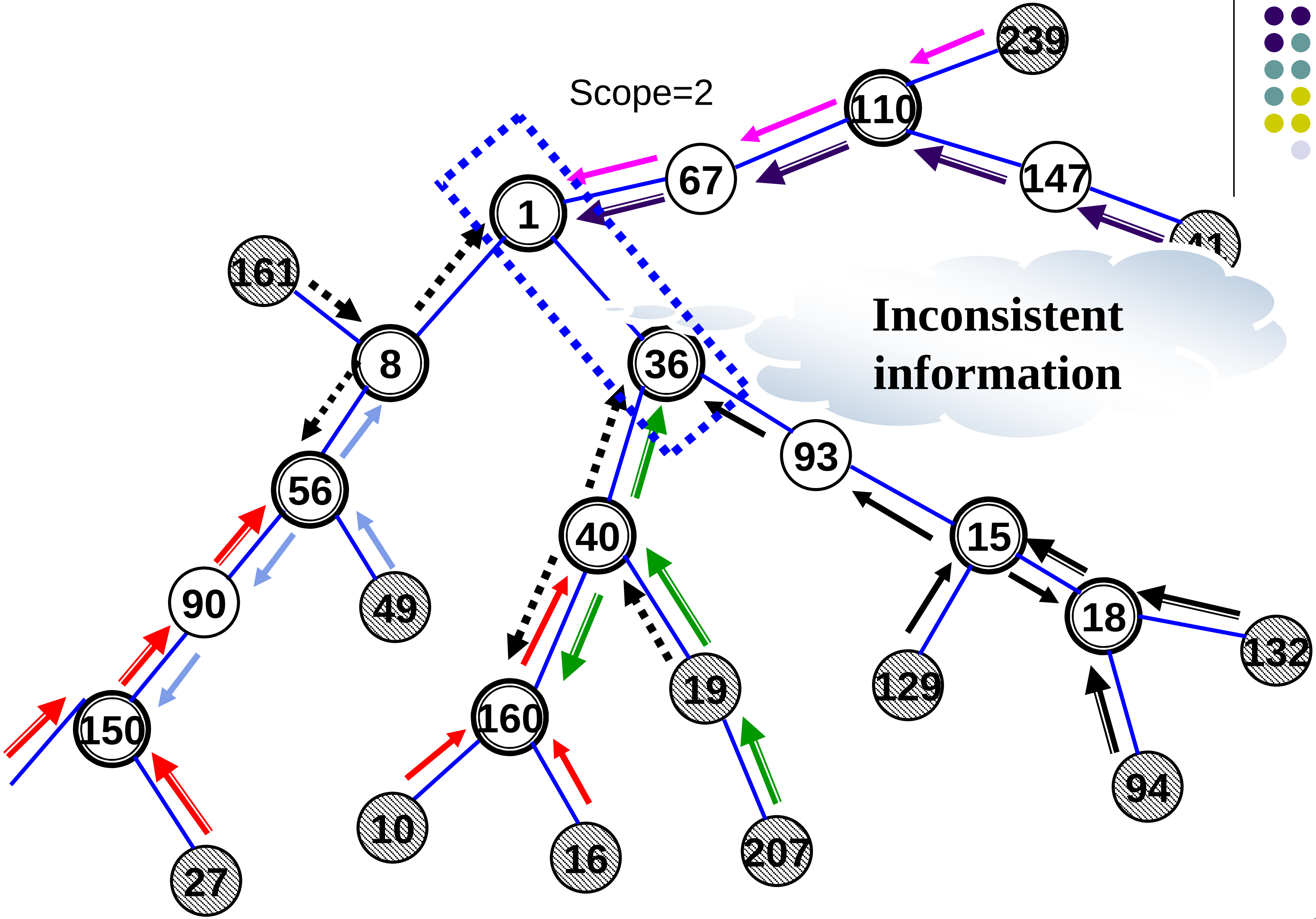
N_i	Id of Neighbors (ID_i)	Lifetime (T_i)	Service Information				
			H_1	H_2	H_3	…	H_m
N_1	ID_1	T_1	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	…	$S_{1,m}$
N_2	ID_2	T_2	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$	…	$S_{2,m}$
□	□	□	□	□	□	…	□
N_n	ID_n	T_n	$S_{n,1}$	$S_{n,2}$	$S_{n,3}$	…	$S_{n,m}$

Service Discovery Scheme

- When a sender node requests services
  - Create a SREQ message
  - In order to enhance the reliability,
    - sender can increase the number of requesting services k to k’ (k’□k)
  - Set number of request services to SREQ{K_No}
- When a SSN receives a SREQ
  - To redistribute the number of request services
    - If total #services in SIT< K_No
    - If total #services in SIT□ K_No
- When an ACH receives a SREQ message,
  - Create a SREP and send SREP message to sender node
  - Check SREQ{K_No} – NumService
    - If SREQ{K_No} >0 :Send SREQ message
    - If SREQ{K_No}□0:Stop forward

Anycast Tree Maintenance Scheme

- When an anycast service rejoins another CH
  - The CH will be an ACH
  - It will send an UNPRUNE message to its parent node and sets its BE_PRUNE flag to FALSE
- When CH receives the UNPRUNE message
  - If its BE_PRUNE flag is TRUE
    - CH will forward an UNPRUNE message to its parent node and set its BE_PRUNE flag to FLASE
  - If its BE_PRUNE flag is FLASE
    - CH will stop forwarding UNPRUNE message
- When an anycast service disjoins from ACH
  - If there are no other anycast services belonging to the ACH
    - The ACH will become a CH
  - If this CH is a leaf node
    - it will send a PRUNE message to its parent node and set its BE_PRUNE flag to TRUE.


Scope=2
Inconsistent information
239
110
67
147
1
41
161
8
36
56
93
40
15
90
49
18
132
150
160
19
129
94
10
16
207
27