

Bootstrapping Chord over MANETs - All Roads Lead to Rome

Wei Ding, *Member, IEEE*, S. S. Iyengar, *Fellow, IEEE*

Abstract — This paper presents a novel approach on bootstrapping Chord over mobile ad hoc networks (MANETs). The approach uses only neighborhood information to build a ring topology in node ID space of structured peer-to-peer (P2P) networks, upon which entire Chord protocol could be put into normal operation without lengthy stabilization. RAN protocol suite is proposed. It includes three patterns: distributed exhaustive, virtual centralized exhaustive, and random. Simulation results show that the distributed exhaustive pattern has optimal overall performance. Once again the superiority of decentralization is proved.

Index Terms — ad hoc networks, bootstrap, Chord, peer-to-peer

I. INTRODUCTION

Research effort and real world application interact in various ways. For P2P networks, the interaction is quite positive — successful business applications aroused public interest and initiated active research, and research in turn brought in better application software. This is a healthy pattern. In the case of MANETs, theoretical research dominated the area for a decade; however, except Bluetooth, no application has been widely applied. This has set off a warning signal in the MANETs community. On the other hand, the inherent homogeneity between P2P systems and MANETs implies promising propability of synergy and transplation.

Majority of research in synergy and transplation deals with problems occurring after a stable P2P network or system has been set up over MANETs. They usually ask questions like how to transfer proven P2P approaches in wired networks to MANETs. [1-5] Little effort has been reflected in publications to explore the mechanism of bootstrapping a P2P network over MANETs. Bootstrapping has largely been circumvented via unrealistic assumptions. Bootstrapping basically do two things: (1) configuration of nodes addresses (and IDs), (2) setting up overlay topology. This paper focuses on the second task. For structured P2P networks, overlay topology lays foundation for other functions like routing, file sharing, looking up, retrieval, and data dissemination. It is one of dominant factors that affect efficiency, robustness, and

feasibility of algorithms.

There are two options for bootstrapping a structured P2P network in wired networks. One is to jumpstart a network from one or few predefined nodes, in which the only way to expand the network is node joining. Many structured P2P systems require the manual creation of a “seed” network in their bootstrapping protocols. A long time may be needed before a topology is ready to use. Another approach is to construct an overlay topology for each connected component. With this option, a P2P system could advance to normal working status immediately after bootstrapping.

Remarkable advance has been achieved in topology generation in wired networks. Present state-of-art generators in wired networks can construct basic network topologies such as line, ring, mesh, star, and tree. General topology generator, which could construct any topology if a mathematical expression of preferred selection criteria could be inputted, is already available. However, to best of our knowledge, no such topology construction protocol in MANETs has been recorded in previous publications. Our proposal probably is the first successful attempt in the specific area.

This paper presents Ring Ad-hoc Network (RAN) protocol suite, a novel solution for bootstrapping structured P2P networks over MANETs and building a ring topology in node ID space. On this ring, Chord [6, 7] could be put into normal operation without lengthy stabilization. The suite includes three patterns: distributed exhaustive, virtual centralized exhaustive, and random. Simulation results show that the distributed exhaustive pattern has optimal overall performance.

II. RELATED WORKS

Cramer and Fuhrmann’s pessimistic verdict [8] on bootstrapping Chord over MANETs, which almost sentenced death to the topic, has been an incentive of this research. We completely disagree with their conclusion. Serious problems could be found in [8]. The whole paper is built upon some unrealistic, far-fetched assumptions. For example, they assume that all nodes can reach a common bootstrap node immediately after they power up. To make it possible, either all nodes in the MANET have to be only one hop away; or every node already has a route to that super node. The most unrealistic assumption states that at the time of power up an ID space ring has been set up and all nodes have already

Wei Ding (corresponding author) is with the University of Maine at Fort Kent. Fort Kent, ME 04743, USA (phone: +1-207-834-7584; fax: 207-834-7577; e-mail: wei.ding@maine.edu).

S. S. Iyengar is with Louisiana State University, Baton Rouge, LA 70803, USA (email: iyengar@csc.lsu.edu).

joined the this ring.

This research has benefited from previous research in T-Man [9], T-Chord [10], and Ring Networks (RN) [11]. T-Man is designed as a general purpose distributed protocol for building and maintaining topology in overlay network. It is intuitive, adaptive, scalable, robust, and flexible. T-Man follows a stepwise refining procedure to increase the quality of the current set of neighbors. Key concept of T-Man is ranking function, which specifies the preference for a node to choose its neighbors in the target topology. A node uses the ranking function to calculate its neighbor set. This simple and elegant abstraction results in an effective algorithm which produces various topologies with efficiency. It gradually evolves the current topology towards the target. The convergence is fast and scalable to network size.

T-Chord efficiently bootstraps Chord from a random unstructured overlay using T-Man. T-Chord creates a perfect Chord topology in $O(\log(N))$ steps. N is network size. It shows optimized message latency. The generated network is immediate operable for Chord. The old pattern of bootstrapping structured P2P system uses a jumpstart node and node joining procedure. Nodes have to be booted one by one in a slow, linear manner. T-Chord completely breaks away from the old pattern.

The RN protocol is an asynchronous message-passing distributed protocol. Nodes use traditional message passing in communication. Peers do not need to be informed of any global network state. The RN protocol is fully distributed. All peers periodically call the Closer Peer Search procedure to find a closer successor candidate. The most serious problem with RN is that it requires a weakly connected initial network called minimal bootstrapping system. It does not specify the scale of the bootstrapping system and how the system is found or configured.

III. TOPOLOGY CONSTRUCTION OVER MANETS

In wired networks like Internet, all nodes are connected. Being neighbor is equivalent to knowing address, which in turn is equivalent to being able to send a message. In MANETs, not any two nodes are guaranteed to be connected. Neighbor relation in MANETs is usually defined by radio range in lowest layer. It has nothing to do with structured P2P ID space or overlay layer. Moreover, the neighborhood relation in wired networks is unidirectional, while in MANETs it is bidirectional and symmetric, since it is defined by radio range. In MANETs, a node's neighbor set is fixed at a given time, while for a node in a wire network, it could have countless variation.

Both P2P over wired networks and P2P over MANETs have proximity concerns, but in MANETs this issue has more serious impact. A hop in MANETs is more costly than in Internet. Hence proximity optimization has more urgent, more realistic significance in MANETs. A P2P network over wired network usually does not include all intermediate nodes on its path on the Network layer. Otherwise the network may have

too many unrelated nodes. In a P2P network over a MANET, the situation is poles apart: all intermediate nodes should be included to secure connectivity on the overlay layer.

IV. RAN PROTOCOL SUITE

A. Introduction

RAN is a protocol to build a ring topology over MANETs. RAN has integrated merits of T-Man and Ring Network. RAN is completely distributed, using only neighborhood information. RAN builds an ideal ring topology for each connected component. Upon this ring, Chord could run immediately without stabilization. Each node first converts its connected component into a *component tree* with itself as the root. At each step, if a node has shorter distance from root than the current successor, it is selected as the new successor. The process repeats till the tree is traversed.

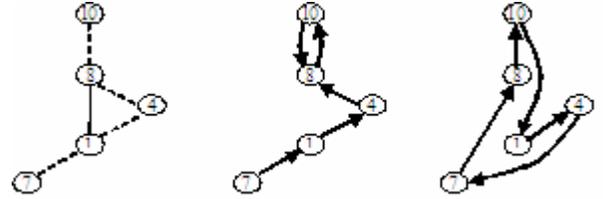


Fig.1 Left shows only neighborhood relation. Middle shows original successor relation. Right is successor relation after running RAN

A component tree is a spanning tree derived from the connected component. We use the assumption of ideal network, which is a network with only one complete connected component. In an ideal network, all nodes have same one node time complexity and same one node message complexity. The network message complexity is the sum of one node message complexities of all nodes. Suppose M is one node message complexity; M_{Net} is network message complexity. For a complete component tree with N nodes, downward degree k , and depth d ,

$$N = \frac{k^{d+1} - 1}{k - 1} \quad (1)$$

$$d = \log_k(kN - N + 1) - 1 \quad (2)$$

$$k^d = kN - N + 1 \quad (3)$$

$$M^{Net} = N \times M \quad (4)$$

B. Three Patterns and Three Options

We designed three patterns for RAN, namely distributed exhaustive pattern, virtual centralized exhaustive pattern, and random pattern. The primary concern is the trade-off between the completeness of ring and complexities of time and message. Two exhaustive patterns exhaustively search closer successor at each level of component tree. The ideal Chord ring is guaranteed, because all node IDs are compared. However, this exhaustion may suffer from high cost. Measures are needed to mitigate the overhead. These two exhaustive patterns are equally excellent in effectiveness. In

virtual centralized exhaustive pattern the searching node serves as a central controller. The distributed exhaustive pattern has better performance for only neighbors exchange messages. The random pattern has its root in Ring Network. We eliminated the minimal bootstrapping system and use a breadth-first search scheme in lieu of it. All nodes in a component tree of the searching node are traversed in a cascading manner.

Three options are also defined to improve the efficiency. Plain option means no additional option. The approximation option could be applied to all three patterns. It loosens the end condition of search. A small fraction of nodes are allowed to be left out of the final rings. In multicast option a node sends message to all direct children by multicasting. It considerably improves time and message complexity. It can not be applied to the random pattern.

C. Distributed Exhaustive Pattern

In the distributed exhaustive pattern, the searching node sends a Closest Successor Request message to each child. Each child concurrently forwards the request message to its children. At following levels, nodes keep forwarding the request message to children at next level until leaf nodes are reached. Then, from leaf nodes up, the closest successor of the root in the subtree is calculated at the root of the subtree and returned to the parent node in a Closest Successor message. The calculation is done by comparing the distance of returned candidates from children. The message complexity of node M satisfies $M = 2(k+1)N - 2k^{d+1} - 2$, where $-2k^{d+1}$ is the number of skipped messages of kd leaf nodes, for they have no children. Let T_d, T_u be downward and upward Time cost, plug in (3), we get

$$\begin{aligned} M &= 4N - 4 \\ M^{Net} &= 4N^2 - 4N \\ T_d &= dk \\ T_u &= d \\ T &= T_d + T_u = d(k+1) \\ T &= (k+1)(\log_k(kN - N + 1) - 1) = O(\log N) \end{aligned}$$

In multicast option,

$$\begin{aligned} M &= (k+3)N - (k+1) \times k^d - 2 \\ M &= (1/k + 3)(N - 1) \\ M^{Net} &= MN = (1/k + 3)(N - 1)N \end{aligned}$$

The Closest Successor Request message needs 1 time unit to move from one level to next, so

$$\begin{aligned} T_d &= T_u = d \\ T &= 2d \\ T &= 2 \log_k(kN - N + 1) - 2 = O(\log N) \end{aligned}$$

D. Virtual Centralized Exhaustive Pattern

In the virtual centralized exhaustive pattern, all direct children nodes of the root form the first level set. Direct children of nodes in first level form the second level, and so on. Current level nodes send their downward children set to root, so root get information of the next level. Then root sets the next level as new current level and repeats the same procedure till leaves are reached. This algorithm is most expensive in terms of overhead. However, it gives the root node tremendous power to control whole process upon distributed network. In real world application, individualized service could be implemented if this pattern is used.

If we count a multi-hop message as one message, both MNet and M are independent of k and depth d ; they only depend on N .

$$\begin{aligned} M &= 2(N - 1) \\ M^{Net} &= 2(N - 1)N \end{aligned}$$

Unlike in the distributed exhaustive pattern, almost all messages have to go through multi-hops. To get precise comparison with the distributed exhaustive pattern, the message per hop M_{hop} should be used. A round trip of the getAllNeighbors message and AllNeighbors message at first level takes $k + 1$ time units. At second level, it is $k^2 + 2$, and so on, till $kd + d$.

$$\begin{aligned} M_{hop} &= 2k + 4k^2 + 6k^3 + \dots + 2dk^d = 2 \sum_{i=1}^d ik^i \\ M_{hop}^{Net} &= NM_{hop} = 2N \sum_{i=1}^d ik^i \\ T &= (k+1) + (k^2 + 2) + (k^3 + 3) + \dots + (k^d + d) \\ T &= N + d(d+1)/2 - 1 \end{aligned}$$

E. Random Pattern

Random pattern is like a RN in MANETs. To it suitable for MANETs, several changes are made to make it completely decentralized and self-organized. The minimal bootstrapping system is eliminated. As its substituent, all nodes in the component are made available using a breadth-first traversal. In worst case, all nodes will be covered. In fact only small number of simulations really used all nodes. In worst case, M consists of M_s (to launch nodes into the search) and M_{search} (to search each node). To launch nodes each node except the root receives a Start message with root included, so $M_s = N - 1$. The second part is the sum of distance to bottom for each node except root,

$$\begin{aligned} M &= M_s + M_{search} \\ &= d(k + k^2 + \dots + k^{d-1}) - [(k + 2k^2 + \dots + (d-1)k^{d-1})] + 2(N - 1) \end{aligned}$$

$$T_{round} = \underbrace{k + k + \dots + k}_d + d$$

In each round, , same as in the

distributed exhaustive pattern.

$$T = T_{round} \times N = (k + 1)dN = O(N \log N)$$

Due to the limit of the page number, detailed algorithms for three patterns and their variants are not given in this paper. Please refer to full paper if interested.

V. SIMULATION

Parameters tested include completeness (percentage of nodes covered in the output ring), time, and numbers of sent and received messages. Resultd with message processing rate at 10 and 1 per time unit are given below.

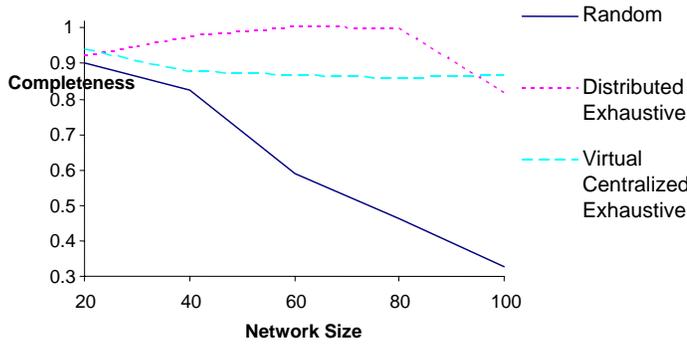


Fig.3 Completeness in approximation option with message processing rate 10

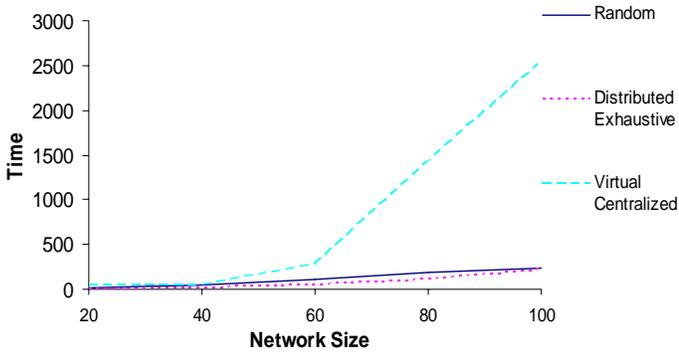


Fig.4 Time in approximation option with message processing rate 10

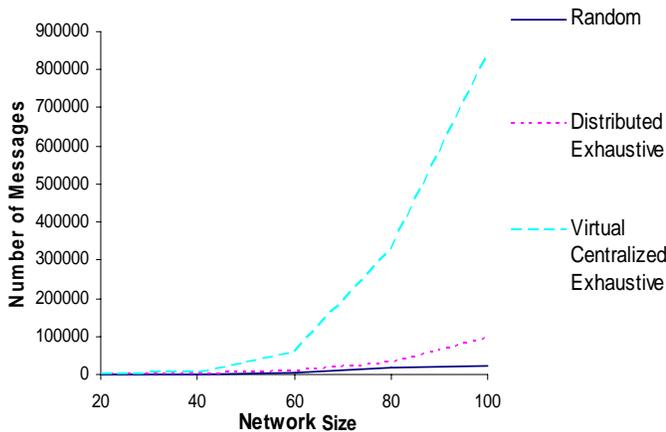


Fig.5 Messages sent in approximation option with message processing rate 10

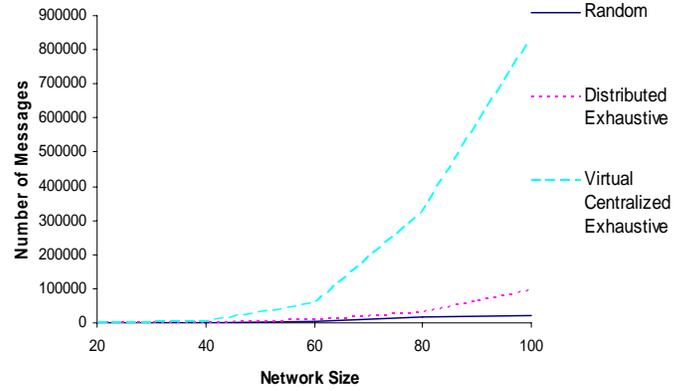


Fig.6 Messages received in approximation option with message rate 10

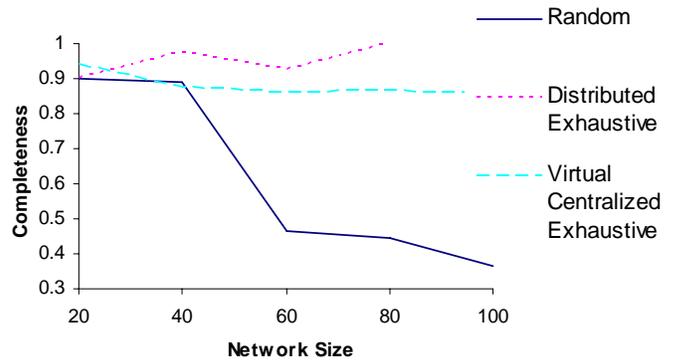


Fig.7 Completeness in approximation option with message processing rate 1

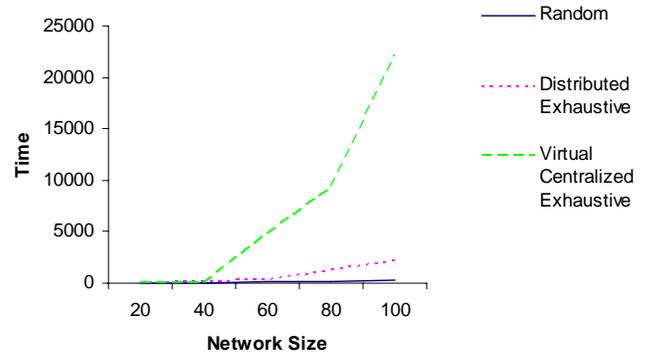


Fig.8 Time complexity in approximation option with message processing rate 1

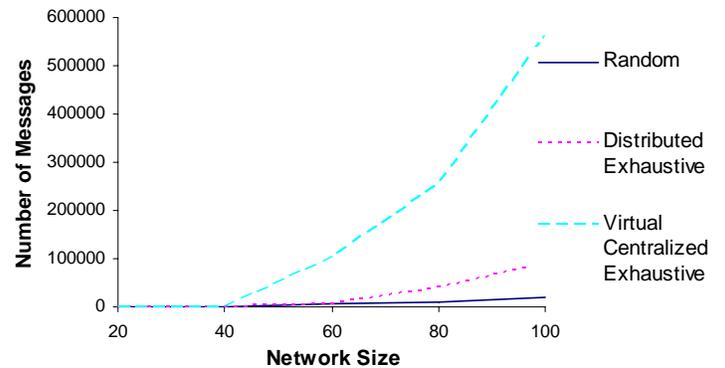


Fig.9 Messages sent in approximation option with message processing rate 1

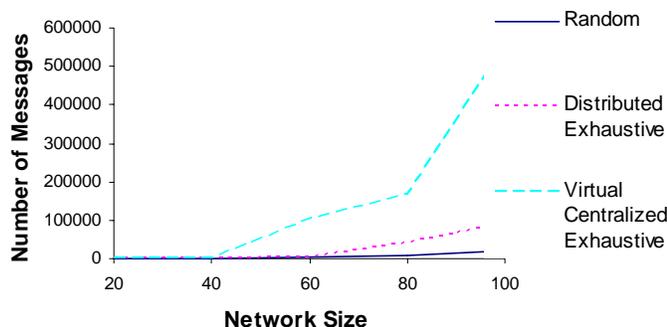


Fig.10 Messages received in approximation option with message processing rate 1

Obviously, the winner is the distributed exhaustive pattern. It shows perfect effectiveness, at the same time and unlike the other two patterns, it has no serious side effect to endanger its advantage. The other two, however, suffer from different fatal problems. For random pattern, it is the effectiveness. For virtual centralized exhaustive pattern, it is efficiency. The benefit of decentralization has shown by the distributed exhaustive pattern. This result also raises a question: could centralization be implemented above decentralized infrastructure?

VI. CONCLUSION

This paper proposes a novel ring construction protocol for building Chord over MANETs. It can be extended to other popular P2P systems like Pastry. This approach inherited successful topology construction methods in P2P over Internet. We believe this approach is the first successful attempt to build ring in P2P ID space over MANETs.

Our RAN protocol suite is an algorithm family for ring topology construction. RAN builds perfect ring in P2P ID space using only simplest multi-hop unicast and multicast communication primitives. No underlying routing protocols are needed. Upon ring generated by RAN, popular P2P networks like Chord could be immediately started and put into normal running without usual lengthy stabilization. RAN family includes a variety of algorithms for ring building. We showed the pros and cons of these algorithms, both in theory and in simulation experiments, and illustrated that the distributed exhaustive pattern is the best in terms of effectiveness and efficiency in time and messages overhead.

REFERENCES

- [1] [HGRW2006] T. Heer, S. Gotz, S. Rieche, and K. Wehrle, "Adapting Distributed Hash Tables for Mobile Ad Hoc Networks," Proceeding of Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops, pp.173 – 178, 2006.
- [2] [LJLQC2004] Sei-yon Lee, Ju-wook Jang, Kyung-Geun Lee, Lan Quan, Tae-kyoung Cho, "A Peer-to-Peer Search Scheme over Mobile Ad hoc Networks," (ISPC) International Scientific-Practical Conference 2004, Institute of Mathematics of National Academy of Sciences (IM NAS, Bishkek, Kyrgyz Republic), 2004.
- [3] [LLS2004] Mei Li, Wang-Chien Lee, Anand Sivasubramaniam, "Efficient peer to peer information sharing over mobile ad hoc

networks," the Second WWW Workshop on Emerging Applications for Wireless and Mobile Access (MobEA'04), New York City, NY, May 2004.

- [4] Y. C. Hu, H. Pucha, and S. M. Das, "Exploiting the Synergy between Peer-to-Peer and Mobile Ad Hoc Networks," in Proceedings of HotOS-IX: Ninth Workshop on Hot Topics in Operating Systems, Lihue, Kauai, Hawaii, May 18-21, 2003.
- [5] Himabindu Pucha, Saumitra M. Das, Y. Charlie Hu, "Ekta: An Efficient DHT Substrate for Distributed Applications in Mobile Ad Hoc Networks," Sixth IEEE Workshop on Mobile Computing Systems and Applications, pp. 163-173, 2004.
- [6] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications," In Proceeding of ACM SIGCOMM 2001, pp. 149-160, San Diego, CA, August 2001.
- [7] Frank Dabek, Emma Brunskill, M. Frans Kaashoek, David Karger, Robert Morris, Ion Stoica, Hari Balakrishnan, "Building Peer-to-Peer Systems with Chord, a Distributed Lookup Service," In the Proceedings of the 8th Workshop on Hot Topics in Operating Systems (HotOS-VIII), Schloss Elmau, Germany, May 2001.
- [8] Curt Cramer and Thomas Fuhrmann, "Bootstrapping Chord in Ad Hoc Networks: Not Going Anywhere for a While," Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06), pp. 168-172, Pisa, Italy, 2006.
- [9] M. Jelasity and O. Babaoglu, "T-Man: Gossip-based overlay topology management," In Engineering Self-Organising Applications (ESOA'05), 2005.
- [10] Alberto Montresor, Márk Jelasity, Ozalp Babaoglu, "Chord on Demand," Fifth IEEE International Conference on Peer-to-Peer Computing (P2P'05), pp. 87-94, 2005.
- [11] Ayman Shaker and Douglas S. Reeves, "Self-Stabilizing Structured Ring Topology P2P Systems," Proceeding of Fifth IEEE International Conference on Peer-to-Peer Computing (P2P'05), pp. 39-46, 2005.