IEEE 6th CAS Symp. on Emerging Technologies: Mobile and Wireless Comm.
Shanghai, China, May 31~June 2, 2004

# Topology-based Capacity Analysis for Ad Hoc Networks with End-to-End Delay Constraints

Junxia ZHANG [2, 1] Winston K.G. SEAH[1]
{stuzjx, winston}@i2r.a-star.edu.sg

[1]Institute for Infocomm Research
Agency for Science Technology and Research
Singapore

[2]Department of Electrical and Computer Engineering
Faculty of Engineering
National University of Singapore

***Abstract*□apacity analysis for an ad hoc network supporting delay-sensitive traffic is addressed in this paper. In contrast, most previous capacity analysis work focused on ad hoc networks carrying delay-tolerant traffic and improved the network capacity at the expense of increased transmission delay. In this paper, ad hoc networks are modeled with adjacency matrices. Based on these adjacency matrices, we design two efficient algorithms, MSDA and CSDA, to derive the capacity of the ad hoc networks carrying delay-sensitive traffic through a sequence of matrix operations. We also design an algorithm to estimate the average hop count of the network.**

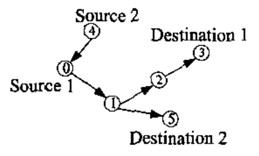
## I. Introduction

In ad hoc networks, the supported services are not only delay-tolerant, such as FTP and email, but also delay-sensitive real-time applications. Much work has been done [1][2][3] to evaluate the capacity of ad hoc networks. However, they focused primarily on the performance issues of delay-tolerant applications under different network models and transmission scenarios and achieved satisfactory network capacity at the expense of increased transmission delay. [4] is one that estimated the maximum number of users that can be supported in the ad hoc network under specific delay constraints.

In this paper, we focus on ad hoc networks carrying delay sensitive traffic. The network capacity is defined as the maximum number of sessions that can be supported in the network under end-to-end delay constraints. A session comprises one hop or several sequential hops without considering whether nodes on it are source, destination, or intermediate nodes (Fig. 1). A session containing ***n*** hops is called as an ***n***-hop session.

In this paper, ad hoc networks are modeled with adjacency matrices. Two matrix-based algorithms are designed for two different scenarios: (i) Matrix Select-Delete Algorithms (MSDA) is designed for the non-channel-sharing scenario where each channel is used by only one session; and (ii) Channel-sharing Select-Delete Algorithm (CSDA) is used to obtain network capacity for the channel sharing scenario. Our algorithms can obtain the capacity of the network with time complexity $O(N^2/k)$, in contrast to the time complexity $O(N/(k+1))^N$ of the Brute-Force Search algorithm [5].

The capacity can serve as a reference or criteria for accepting new communication requests, to ensure that, any of the source-destination pairs containing these sessions will meet end-to-end delay constraints.



| Path 1: 0->1->2->3 | |
|---|---|
| hop count | sessions |
| one-hop | 0->1 1->2 2->3 |
| two-hop | 0->1->2 1->2->3 |
| three-hop | 0->1->2->3 |

Fig. 1. Paths and the Sessions (A session can be shared by two or more paths. For example, one-hop session 0->1 can be shared by path 1 and path 2.)

The paper is organized as follows. Next section describes the mathematical model for the ad hoc networks. In section III, the design and implementation of the Matrix Select-Delete Algorithm (MSDA) is described. Section IV introduces Average Hop Count Algorithm. In section V, we present Channel-sharing Select-Delete Algorithm (CSDA) for capacity estimation with channel sharing. Finally, we state the conclusions and discuss some future work in section VI.

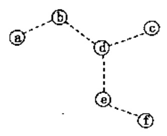
## II. Mathematical Model

The topology of an ad hoc network is modeled by an undirected graph ***G(V,A)***. ***V*** denotes the node set in the network and ***A*** is an adjacency matrix that describes the topology of the network.

An *adjacency matrix* of a graph is a {0,1} matrix in which the $ij^{th}$ entry is 1 if there is an edge between ***node i*** and ***node j*** and all other entries of the matrix are zero [6]. In our case, "1" denotes two corresponding nodes are in the transmission range of each other and "0" denotes they are not (Fig. 2).

In Fig.2, matrix *A* is called as one-hop adjacency matrix because it only contains one-hop paths. We extend one-hop adjacency matrix to multi-hop matrices according to the following proposition:

Proposition [6]: Let G=(V,E) be a graph with vertex set $V=\{v_1, v_2, \dots v_n\}$ and let $A^k$ denote the *k*-th power of the adjacency matrix (The matrices are multiplied as is usual in linear algebra, i.e. if we put B = $A^2$, we have $b_{ij} = \sum_{k=1}^{n} a_{ik} a_{kj}$ ). Let $a_{ij}^{(k)}$ denote the element of the matrix $A^k$ at position (i,j). Then $a_{ij}^{(k)}$ is the number of walks of length exactly *k* from the vertex $v_i$ in the graph G.



$$A=\begin{array}{c|cccccc} & a & b & c & d & e & f \\ \hline a & 1 & 1 & 0 & 0 & 0 & 0 \\ b & 1 & 1 & 0 & 1 & 0 & 0 \\ c & 0 & 0 & 1 & 1 & 0 & 0 \\ d & 0 & 1 & 1 & 1 & 1 & 0 \\ e & 0 & 0 & 0 & 1 & 1 & 1 \\ f & 0 & 0 & 0 & 0 & 1 & 1 \end{array}$$

Fig. 2. The network topology and its Adjacency Matrix (Dashed line denotes two nodes are within the transmission range of each other.)

We define a special process to deal with the matrix multiplication – $A^i(n,n)\times A(n,n)$:

*Exact Multiplication (EM)* (denoted by$[x\times x\times\cdots\times x]^{*}$) is defined as follows: If $A^i(u,v)=0$ and $A^{i+1}(u,v)>0$, we let $A^{i+1}(u,v)$ equal to $i$+1; if $A^i(u,v)>0$, $A^{i+1}(u,v)$ is set the same value as $A^i(u,v)$. The matrix obtained finally shows the exact shortest hop count between any arbitrary pair of nodes.

Fig. 3(A) shows the 2-hop adjacency matrix $A^2(6,6)$ obtained by using Exact Multiplication on $A(6,6)\times A(6,6)$. It lists all the node pairs which can reach each other within two hops. Similarly, we can get from $A^3(6,6)$ (Fig. 3(B)) to $A^n(6,6)$ where $\boldsymbol{n}$ is the network diameter.

## III. Capacity Under Non-Channel-Sharing

We assume that every Source-Destination pair in the ad hoc network communicates through a common broadcast channel using omni-directional antennas with the same transmission range. As in [4], we assume the packets travel one hop in each time slot. Thus, the end-to-end delay can be measured as the number of slots required for a packet to be sent successfully.

### A. Matrix Select-Delete Algorithm (MSDA)

In this section, we focus on the non-channel-sharing scenario, that is, each channel is used by only one session. Each session belongs to only one path so that if each path is seen as a same hop session, the number of sessions equals to the number of paths. Based on the one-hop and multi-hop adjacency matrices, we propose the Matrix Select-Delete Algorithm (MSDA), as shown below. The algorithm comprises a series of selection iterations. Rules (1) and (2) guarantee the maximum number of available nodes remain after each iteration in order to obtain maximum number of paths. Rule (3) is designed according to the transmission property of the wireless ad hoc networks (Fig. 4).

$$A^2=\begin{array}{c|cccccc} & a & b & c & d & e & f \\ \hline a & 1 & 1 & 0 & 2 & 0 & 0 \\ b & 1 & 1 & 2 & 1 & 2 & 0 \\ c & 0 & 2 & 1 & 1 & 2 & 0 \\ d & 2 & 1 & 1 & 1 & 1 & 2 \\ e & 0 & 2 & 2 & 1 & 1 & 1 \\ f & 0 & 0 & 0 & 2 & 1 & 1 \end{array} \qquad A^3=\begin{array}{c|cccccc} & a & b & c & d & e & f \\ \hline a & 1 & 1 & 3 & 2 & 3 & 0 \\ b & 1 & 1 & 2 & 1 & 2 & 3 \\ c & 3 & 2 & 1 & 1 & 2 & 3 \\ d & 2 & 1 & 1 & 1 & 1 & 2 \\ e & 3 & 2 & 2 & 1 & 1 & 1 \\ f & 0 & 3 & 3 & 2 & 1 & 1 \end{array}$$

(A) (B)

Fig. 3. 2-hop (A) and 3 -hop (B) adjacency matrix

```
Begin [Matrix Select-Delete Algorithm (MSDA)]
  Input number of nodes ($n$) and one-hop adjacency matrix ($A(n,n)$)
  Input hop count of the available paths ($k$)
  Compute $B(n,n)=\Big[\underbrace{A(n,n)\times A(n,n)\times\cdots\cdots\times A(n,n)}_{k}\Big]^{*}$
  Store all the paths in PathSet;
  SelectedPaths := NULL;
  While (PathSet <> NULL)
  { (1)source := select the node with the fewest one-hop neighbors;
    (2)dest := select the node, which is one of k-hop neighbors of
       the source, and has the fewest one-hop neighbors;
    AddPath(SelectedPaths, source, dest): Add path
       originating from source to dest, to SelectedPaths;
    (3)B(n,n) := delete all the columns and rows of the source
       destination, relay nodes and their one-hop neighbors;
    PathSet := delete all corresponding paths according to the
       deletion of B(n,n) from PathSet;
  }
  Output(SelectedPaths);
End
```

### B. Algorithm validation

We compare the results of MSDA with those of Brute-Force Search by choosing the shortest paths under the same conditions. A Brute-Force Search algorithm systematically enumerates every possible solution of a problem until an optimal solution is found, or all possible solutions have been exhausted.

Ten scenarios are used for simulations. In $i_{th}$ scenario, all the valid shortest paths are $\boldsymbol{i}$ hops ($\boldsymbol{i}=1,2\ldots10$).

In Fig. 5(A), the number of nodes of the ad hoc network is 26 and the average number of neighbors is 3. The simulation results are shown in Fig. 5(B).

In Fig. 6(A), the number of nodes is 40 and the average number of neighbors per node is 3. The simulation results are illustrated in Fig. 6(B). From the Fig. 5(B) and Fig. 8(B), we can see that MSDA can obtain the results close to that of Brute-Force Search with a deviation of one. While the time consumed by two algotithm is different. Given $\boldsymbol{N}$ nodes and $\boldsymbol{k}$ hops between all sources and destinations. The time complexity of Brute-Force Search is $O[N/(k+1)]^N$ and that of MSDA is $O(N^2/k)$. Therefore, the time required by MSDA is

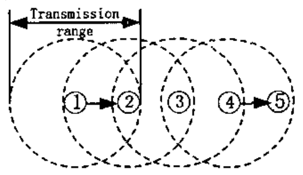
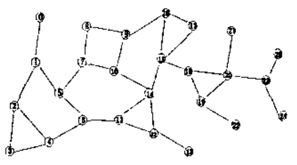
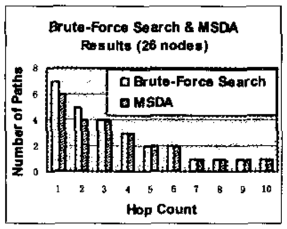
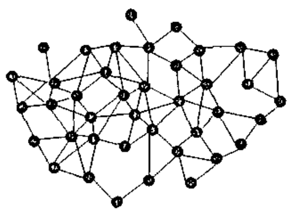


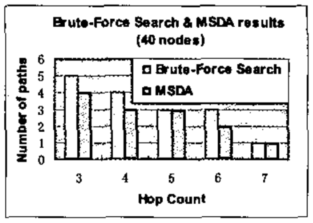
Fig. 4. Transmission property (When node 1 is transmitting, the node 4 can transmit simultaneously while node 2, 3 cannot due to interference.)





(A) (B)
Fig. 5. Simulation Topology (A) and Brùte-Force Search vs MSDA (B) for 26-node network





(A) (B)
Fig. 6. Simulation Topology (A) and Brute-Force Search vs MSDA (B) for 40-node network

much shorter than Brute-Force Search. Due to the mobility property, resulting in the topology of the ad hoc network changing frequently, our algorithm can therefore quickly estimate the current network capacity, making it feasible for real-time capacity estimation.

## IV. Capacity Computation Under Channel Sharing

In this section, we focus on scenario that two or more paths share common channels. The capacity here is the maximum number of one-hop sessions with channel bandwidth constraints and end-to-end delay constraints, inasmuch as packets are assumed to be transmitted one hop in one time slot.

### *A. Average hop count algorithm*

Average hop count (AHC) is an indicator for the statistical situation of the paths in the network.

$$Average\ \ hop\ \ count = \frac{\sum_{\substack{All\ possible\\ communication}} hop\ \ count\ \ of\ \ shortest\ \ path}{number\ \ of\ \ source-destination\ \ pairs} \tag{1}$$

We propose an algorithm based on adjacency matrices of the network and (1). Assuming the number of nodes in the ad hoc network is ***n*** and the largest hop count of shortest path is ***l***, then:

```
Begin
  Input dc (end-to-end delay constraint), A(n,n)
  k = min[dc,l]
  Compute $B(n,n) = [A^k(n,n)]^* = \left\lceil \underbrace{A(n,n)\times A(n,n)\times\cdots\cdots\times A(n,n)}_{k} \right\rceil^*$
  For all value i in the matrix B(n,n), except items on diagonal
      Compute $sum = \sum_{i\in B(n,n)} i - n$
  Compute $AHC = \left\lceil \frac{sum}{n^2-n-n_0} \right\rceil = \left\lceil \frac{\sum_{i\in B(n,n)} i - n}{n^2-n-n_0} \right\rceil$          (2)
End
```

$n_0$ is the number of the items with value 0 in the matrix B(n,n), and $\lceil x \rceil$ gives the smallest integer greater than or equal to ***x***.

### *B. Capacity Estimation*

Based on the notion of average hop count, we extend our algorithm to more pervasive scenarios. First, we define the following:

Number of nodes is *N* and a node's transmission radius is ***r***.
The bandwidth of the channel is $BW_{node}$.
The bandwidth needed by a transmission is $BW_{transt}$.
End-to-End Delay constraint is $D_E$.
Hop-by-Hop Delay constraint is $D_H$.
Hop-by-Hop delay constraint is expressed as:

$$D_H = \frac{D_E}{AHC} = \frac{D_E}{\dfrac{\sum_{i\in B(n,n)} i - n}{n^2-n-n_0}} = \frac{D_E(n^2-n-n_0)}{\sum_{i\in B(n,n)} i - n} \tag{3}$$

The number of one-hop sessions that the channel can support is $\lfloor BW_{node}/BW_{packet} \rfloor$. In addition, under end-to-end delay constraints, the number of one-hop sessions that share the same channel is: $N_s = \min\left\lceil \left\lfloor \frac{BW_{node}}{BW_{packet}} \right\rfloor, \frac{D_E(n^2-n-n_0)}{\sum_{i\in B(n,n)} i - n} \right\rceil$ (4).

Based on the above, we propose the Channel-sharing Select-Delete Algorithm (CSDA), as shown below.

### *C. Simulations*

We present some simulation results based on two 26-node network topologies. All flows have the same transmission rate of 750 Kbps, and the network channel bandwidth is 2 Mbps. The average hop count of the ad hoc network in Fig. 7(A) is 4, and the number of one-hop sessions without channel sharing is 6 according to the MSDA (Fig. 5(B)).

In the simulation, we randomly add flows with shortest path hop count smaller than the end-to-end delay constraint into the network, one by one until the network is saturated. Each new flow is added on the condition that it does not cause any on-going flow to violate the end-to-end delay constraint. The number of one-hop sessions is calculated according to Fig. 4, where two one-hop sessions is on the flow from node 1 to node 5. Generally, the number of one-hop sessions on an ***n***-

hop path is $\lceil n/3 \rceil$. Simulation results are shown in Fig. 7(B) while Fig. 8(B) shows the simulation results of another ad hoc network with topology shown in Fig. 8(A).

```
Begin [Channel-sharing Select-Delete Algorithm (CSDA)]
  Input number of nodes ($n$) and adjacency matrix ($A(n,n)$)
  Select paths from $A(n,n)$, and store all the paths in PathSet;
  SelectedPaths := NULL;
  While (PathSet <> NULL)
  {
    source := select the node with the fewest one-hop neighbors;
    dest := select the node, which is the one-hop neighbor of the
            source, and has the fewest one-hop neighbors;
    AddPath(SelectedPaths, source, dest): Add path originating
            from source to dest, to SelectedPaths;
    $A(n,n)$ := delete all the columns and rows of the source
            destination and their one-hop neighbors;
    PathSet := delete all corresponding paths according to the
            deletion of $A(n,n)$ from PathSet;
  }
  count := the number of elements in set SelectedPaths;
  Compute $N_{session} = \left\lfloor count \times \min\left[ \left\lfloor \frac{BW_{node}}{BW_{packet}} \right\rfloor , \frac{D_E(n^2 - n - n_0)}{\sum_{i \in B(n,n)} i - n} \right] \right\rfloor$    (5);
  Output ($N_{session}$);
End
```

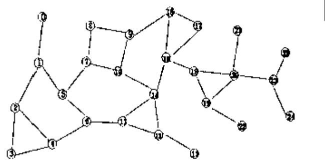
$\lfloor x \rfloor$ denotes the largest integer that does not exceed $x$.

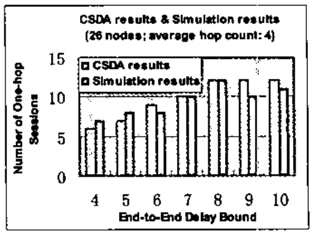
In Fig. 7(B) and Fig. 8(B), the network capacity obtained from CSDA approximates that of simulations with a deviation of one except for the case of the topology in Fig. 7(B) when the end-to-end delay constrain is 9 hops.

From above results, we can see that when the end-to-end delay bound is equal to or larger than 8 hops, the number of one-hop sessions the network can support is similar, because the number of one-hop sessions sharing one channel is also bounded by the limited bandwidth of the channel. Therefore, even though the end-to-end delay constraint is allowed to increase, the number of simultaneously existing one-hop sessions in the network would not change.

## V. Conclusions And Future Work

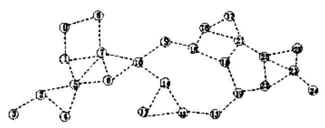
In this paper, we proposed two algorithms Matrix Select-Delete Algorithm (MSDA) and Channel-sharing Select-Delete Algorithm (CSDA) to obtain the network capacity for the non-channel-sharing and channel-sharing scenario, respectively. We also proposed an average hop count algorithm by calculating the probabilities of each possible shortest path hop count. MSDA and CSDA estimate the maximum number of sessions that can exist simultaneously in the network, which is an important network capacity indicator. In contrast with Brute-Force Search methods, our algorithms are much more efficient. From (5), we can see that the capacity of an ad hoc network is restricted by the bandwidth of channels as well as the end-to-end delay constraint. When the end-to-end delay constraint is small, it limits the number of sessions sharing the same channel. By increasing the end-to-end delay constraint, the network capacity would be limited mainly by the bandwidth of the channel. Therefore, the capacity cannot increase unlimitedly. From (5), we can also see that the smaller the average hop count of the flows existing in the network, the more simultaneous one-hop sessions can be supported.

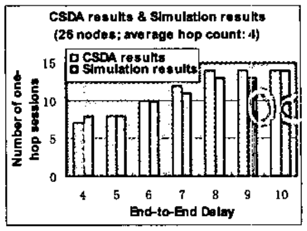



CSDA results & Simulation results
(26 nodes; average hop count: 4)
CSDA results
Simulation results
Number of One-hop Sessions
15
10
5
0
4
5
6
7
8
9
10
End-to-End Delay Bound

(A) (B)

Fig. 7. Network topology (A) and CSDA performance (B)




CSDA results & Simulation results
(26 nodes; average hop count: 4)
CSDA results
Simulation results
Number of one-hop sessions
15
10
5
0
4
5
6
7
8
9
10
End-to-End Delay

(A) (B)

Fig.8. Network topology (A) and CSDA performance (B)

In this paper, we have not considered the interference among the nodes. Thus, the capacity calculated by MSDA or CSDA is the upper bound on capacity achievable in real ad hoc networks. Similarly, the delays in nodes due to contention for shared paths are to be addressed in our ongoing research and future work.

## References

[1] P. Gupta and P.R. Kumar, "The capacity of wireless networks", IEEE Transactions on Information Theory, vol. 46, no.2, March 2000.

[2] S. Toumpis and A. Goldsmith, "Ad hoc network capacity", Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers, Vol.2, Page(s): 1265 -1269, 2000.

[3] J. Li, Z. J. Haas, and M. Sheng, "Capacity evaluation of multi-channel multi-hop ad hoc networks", IEEE International Conference on Personal Wireless Communications, ICPWC 2002, New Delhi, India, Dec 2002.

[4] C. Comaniciu and H.V. Poor, "On the capacity of mobile ad hoc networks with delay constraints", IEEE CAS Workshop on wireless communications and networking, Pasadena, CA, September, 2002.

[5] Anany V. Levitin, "Introduction to the design & analysis of algorithms", Villanova University , ©2003 ISBN: 0-201-74395-7, Page(s): 97-120, 113-119.

[6] Jiří Matoušek and Jaroslav Nešetřil, "Invitation to discrete mathematics", Clarendon Press, Oxford. 1998, Page(s): 109-110.