

Collision Detection and the Design of Fair and Stable MAC Scheme for Wireless Ad Hoc Networks

Yongkang Xiao, Rong Xiao, Bo Sun

School of Information Science and Technology, Beijing Normal University, Beijing, China
Email: xiaoyk@bnu.edu.cn

Received July, 2013

ABSTRACT

Fairness and stability guarantee among TCP flows is very stubborn in wireless ad hoc networks. There is not a MAC protocol that can fulfill this requirement until now. In this paper, we firstly reveal the in-depth causes of the severe TCP unfairness and instability problems in IEEE 802.11-based multihop networks. Then we utilize the collision detection mechanism of the IEEE 802.11 protocol which is often ignored by most of the people to design a novel collision detection mechanism-based MAC (CDMB-MAC) scheme to solve the short-term and long-term fairness and stability issues while providing a good aggregate throughput in many topologies.

Keywords: Ad hoc Networks; IEEE 802.11 MAC Protocol; TCP; Fair; Stable; Collision Detection Mechanism

1. Introduction

Currently, the IEEE 802.11 distributed coordination function (DCF) [1] has been the *de facto* access standard for wireless ad hoc networks. However, Reference [2,3] have shown TCP flows over IEEE 802.11 MAC obtain extreme unfairness, instability and incompatibility problems in multihop ad hoc networks. And they have also shown that all these issues are rooted in the MAC layer. Therefore, researches of how to improve the performance of IEEE 802.11 (e.g., the fairness issue) have been receiving an increasing attention [4-6]. Unfortunately, all of these solutions only aim at the improvement of long-term fairness property, i.e., the average throughput share among TCP flows during the whole simulation interval. They do not consider the fairness property in the short time scale, e.g. second-level. More importantly, there are few papers dealing with the stability of TCP traffic currently.

In this article, we try to find a MAC scheme that can provide a fair and stable communication for TCP flows while achieving high aggregate throughput. Note that if we can acquire short-term fairness and stability performance, we naturally resolve the incompatibility problem among TCP flows. In order to achieve this aim, we firstly introduce the collision detection mechanism of IEEE 802.11 in detail. Then by analyzing throughput results of TCP flows in a classic topology, we provide a novel collision detection mechanism-based MAC (CDMB-MAC) scheme to resolve TCP fairness and stability performance while maintaining a good aggregate throughput.

The main advantage of CDMB-MAC is that it is easier to implement than other methods since it only requires minimal modifications to the IEEE 802.11 protocol. To the best of our knowledge, this is the first attempt to address the fairness and stability problem of TCP simultaneously in the design of MAC protocol for wireless ad hoc networks.

2. Collision Detection Mechanism of 802.11

Before proceeding further we point out that the analysis and simulation results reported in this article are based on the NS2 network simulator (version 2.32) [7]. With a few exceptions, we keep most of simulation parameters as the default setting of NS2. And we only research the Request To Send/Clear To Send (RTS/CTS) scheme of IEEE 802.11 DCF and indiscriminately use the term "IEEE 802.11" and the term "RTS/CTS scheme".

As a contention-based MAC scheme, the contention detection mechanism plays an important role for IEEE 802.11. When a node receives a packet, the power level at which the packet was received is compared with two different values: the carrier sense threshold and the receive threshold. If the power level falls below the carrier sense threshold, the packet is discarded as noise. If the received power level is above the carrier sense threshold but below the receive threshold, the packet is marked as a packet in error before being passed to the MAC layer. Otherwise, the packet is simply handed up to the MAC layer.

Once the MAC layer receives a packet, it checks to

insure that its receive state is presently idle. If the receiver is not idle, one of two things can happen. If the power rate of the packet already being received (carrier) and the received power level of the new packet (interference), $CINR$, is greater than $CPTresh$ (the default value is 10.0 dB in NS2), we assume capture, discard the new packet, and allow the receiving interface to continue with its current receive operation. Otherwise, a collision occurs and both packets are dropped. In other words, whether a collision occurs at the receiver of the MAC layer depends on the value of $CINR$ [7,10].

3. Performance Problem of IEEE 802.11

Consider a classic topology with four nodes (0 through 3) as shown in **Figure 1**. All nodes communicate with a bandwidth of 2 Mbps, and the transmission range is equal to 250 m. According to the IEEE 802.11 protocol implementation in the NS2 simulation software, the interfering range (sensing range) is 2.2 times the size of the communication range, i.e. 550 m. The distance between any two neighboring nodes is equal to 200 m. We set up two TCP-Reno sessions as shown in **Figure 1**. The first session is from node 0 to 1 and the second from 3 to 2. Each TCP flow is an FTP session transferring a large size file and runs for 300 s. The TCP packet size is equal to 1460 bytes and the TCP maximum window size window is 1. The routing protocol is the dynamic source routing (DSR) protocol [8].

Figure 2 shows the throughput results of the two flows. The plotted values of the throughput are measured over each 1.0 second interval. We count the successively received TCP packets in each 1.0 interval and transfer it into the throughput in that interval. We can observe that two TCP sessions suffer from the serious unfairness problem. In most of simulation lifetime, the throughput of the second session is zero. The aggregate throughput of these two TCP connections belongs completely to the first session around 746.362 Kbps.

Note that node 2 is in the transmission range of node 1, it can receive and decode the CTS frame correctly, and the virtual carrier sensing mechanism of IEEE 802.11 makes node 2 know when the current transmission between 0 and 1 will end. However, the hidden node 3 can only sense the CTS frame, but it can not decode the frame, because the distance between 1 and 3 is equal to 400 m which is greater than the transmission radius (250 m). Now if node 3 has data to send to 2, it will defer a distributed interframe space (DIFS) interval and then generate a random backoff period before transmitting. When the backoff timer reaches zero, node 1 is silent while receiving the TCP-Data from 0, however node 3 cannot sense the Data. Therefore, node 3 believes the medium is idle and transmits an RTS to 2. Unfortunately,

a collision occurs at ode 2 and it cannot reply a CTS to 3, so that the latter times out waiting for a CTS frame and adopts the binary exponential backoff (BEB) algorithm to compute a new random backoff time to retransmit the RTS frame. Since the BEB algorithm assigns a larger backoff counter to the failed node, node 3 cannot compete the channel immediately even when the transmission between 0 and 1 finishes. After failing seven times to receive CTS from node 2, node 3 quits and reports a link breakage to its upper layer. Then a route failure event occurs. Then the TCP session has to wait before a route becomes available again. On the other hand, node 0 does not know that node 3 wants to communicate with 2 and starts a new transmission immediately after a successful one.

Here, it is very interesting why node 3's RTS frames do not collide with TCP-Data transmitted by node 0 at node 1. Although the reason is often ignored by most of the people, it is very important to the improvement of TCP performance. According to the two-ray ground reflection model [7], the received power at distance d is predicted by

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L} \quad (1)$$

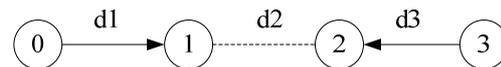


Figure 1. A classic topology with two TCP connections.

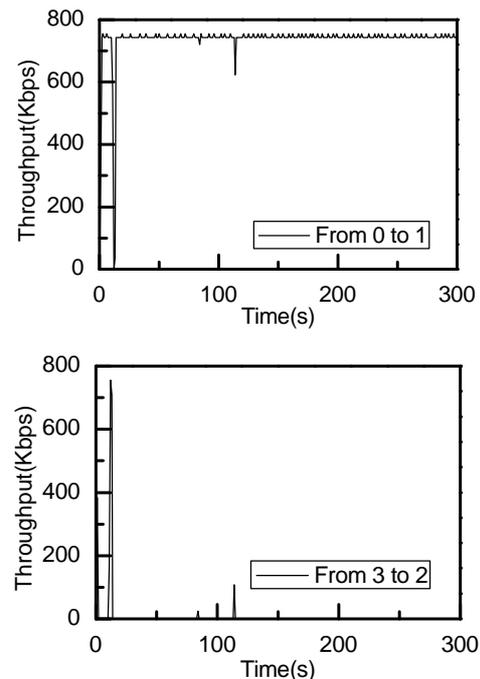


Figure 2. TCP throughput in the IEEE 802.11-based network, $d_1 = d_2 = d_3 = 200\text{m}$, $window = 1$.

where P_t is the transmitted signal power. G_t and G_r are the antenna gains of the transmitter and the receiver respectively. L ($L \geq 1$) is the system loss. h_t and h_r are the heights of the transmit and receive antennas respectively. We assume that each node is equipped with the same wireless radio, therefore $CINR$ is only dependant on the distance of the receiver to two senders. For example, the $CINR$ at node 1 in **Figure 1** is

$$CINR_1 = \frac{P_{01}}{P_{31}} = \left(\frac{d_2 + d_3}{d_1} \right)^4 = 16 > CPTresh = 10 \quad (2)$$

where P_{01} and P_{31} is the received power of signal and interference at node 1 transmitted by node 0 and 3 respectively. According to the collision detection mechanism, if node 1 is receiving a Data from 0, a new arriving RTS frame from node 3 can only defer the channel access of 1, but does not collide with the TCP-Data from 0. However, the default value of RTS retry count (RRC) is so small that node 3 reports a link breakage to its upper layer too early which causes a serial of problems mentioned above. Therefore, the deferece occurring at node 1 does not have an effect on node 0, and the latter starts a new transmission as soon as it receives the ACK frame. An intuition here is that a large value of RRC should be able to effectively suppress the channel access of node 0 and 1 and at the same time give a high opportunity to node 3 to acquire the channel successfully after a MAC layer transmission of the first session finishes.

4. Collision Detection Mechanism-Based Scheme

The CDMB-MAC scheme can be summarized as follows: When the $CINR$ value is greater than the $CPTresh$ value, a node transmits with a probability p if the channel is idle for a period of time equal to a DIFS. With a probability $q = 1-p$, it backoffs with a fixed time interval T . When the backoff timer timeouts, and if the channel is also idle, it either transmits or defers again with the same probabilities p and q . This process is repeated until either the frame has been transmitted or the maximum RTS retry count RRC is reached. In the latter case, the MAC layer of the unlucky node will drop its transmitting packet and report a link breakage to its upper layer. If the node initially senses the channel busy, it waits until the medium becomes idle without interruption for a DIFS, then applies the above algorithm. Note that when a node retransmits an RTS frame, the backoff window size is still equal to T . For CDMB-MAC, three important parameters need be designed carefully, i.e., maximum RTS retry count RRC , backoff window size T and transmission probability p . According to the simulation results, we set RRC as 200, T as the minimum contention window of IEEE 802.11 MAC protocol, that is $T = CW_{min}$ (31

slot time) [7], and p as 0.4.

For CDMB-MAC, each node determines its RTS transmission or waits only according to its own channel state. This is a main advantage over the traditional p-persistent CSMA algorithm in wireline networks, because CDMB-MAC does not require to divide time into discrete intervals and thus does not need a synchronization scheme to make nodes agree on slot boundaries.

5. Performance Evaluation

The goal of the following simulation is to evaluate the performance of the CDMB-MAC scheme in various scenarios. The maximum window size of TCP, $window$, is equal to 8 in the next simulations.

5.1. Classic Topology

1) Case 1: $d_1 = d_2 = d_3 = 200m$: In this scenario, the distance between node 0 and 3 is 600 m and is greater than the carrier sense range, therefore the two nodes are fully independent. The main problem with IEEE 802.11 is the serious long-term unfairness and incompatibility problems as shown in **Figure 2**.

Figure 3 is TCP throughput of the two flows in the CDMB-MAC-based network. In the 300 s lifetime of the two flows, there is not one time when the throughput reaches zero. The aggregate throughput in **Figure 2** (752.522 Kbps) is greater than that in **Figure 3** (680.042 Kbps), however the stability and short-term fairness performance of the latter is far better than that of the former. And two simultaneous TCP traffics can coexist in the

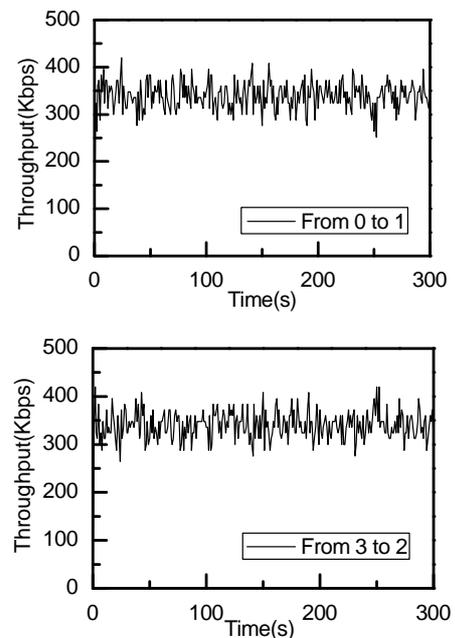


Figure 3. Throughput of two TCP flows in the CDMB-MAC network, $d_1 = d_2 = d_3 = 200m$.

network at the same time. It is well-known that there is a fundamental conflict between achieving flow fairness and maximizing overall throughput. We must have a trade-off between aggregate throughput and fairness. Therefore, this result is really what we need.

From the simulation trace file of NS2 which is not presented here, we can find that node 0 has transmitted 8493 different RTS frames successfully. About 52% frames are transmitted without any retransmission; about 40% frames are transmitted with a retry count less than 10; and about 2% frames need be retransmitted above 15 times. A similar case also occurs at node 3. The biggest RTS retry count for nodes 0 and 3 are 29 and 34 respectively. Obviously if we set $RRC = 7$ (the default setting of IEEE 802.11), CDMB-MAC will also results in a serious fairness and incompatibility issue. A large RTS retry attempts will give two flows a big opportunity to access the channel, so that they can simultaneously share the channel stably and fairly.

2) Case 2: $d_1 = d_3 = 200$ m, $d_2 = 155$ m: In this scenario, the distance between node 1 and 2, d_2 , is reduced to 155 m. **Figure 4** shows TCP throughput of the two flows in the CDMB-MAC network. From the long-term fairness point of view, the two flows can fairly share the channel. This result is better than that in the IEEE 802.11-base network where one flow monopolizes the channel and the other is starved. However, in the 300 s lifetime of two connections, they cannot share the channel simultaneously and there are 66 times when the aggregate throughput reaches zero. Therefore, from the short-term fairness and stability points of view, this result is not really what we need.

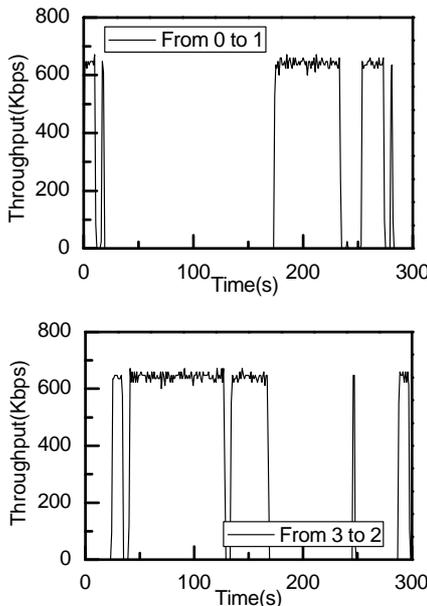


Figure 4. Throughput of two TCP flows in the CDMB-MAC network, $d_1 = d_3 = 200$ m, $d_2 = 155$ m.

The reason can be explained as follows. In this scenario, the distance between node 0 and 3 is 555 m which is greater than the carrier sense range, therefore node 3 cannot sense the frame transmitted by 0, either. From Equation (2), the CINR at both node 1 and 2 are equal to 9.926 (<10). When node 1 is receiving the TCP-Data from 0, the new or retransmitted RTS frame from node 3 would collide with the TCP-Data frame, therefore node 1 must drop two frames. This case also occurs at node 2. These collisions and drops degrade seriously TCP performance.

Fortunately, we can utilize the node mobility to improve the TCP performance. As we all known, node mobility is an important characteristic of ad hoc networks. However, arbitrary movement does not benefit our CDMB-MAC scheme. Since the transmitters need to communicate with the receiver, they should move cooperatively so that CINR of the receiver can be greater than $CPTresh$. Here if the receivers can move only one meter towards the transmitters, we can also achieve an ideal result similar to that shown in **Figure 3**.

3) Case 3: $d_1 = 199$ m, $d_2 = 155$ m, $d_3 = 200$ m: From Equation (2), the CINR at node 1 is equal to 10.1275 (>10), and that in node 2 is equal to 9.8151 (<10) in this scenario. **Figure 5** shows TCP throughput of the two flows in the CDMB-MAC network. In the 300 s lifetime of the two flows, there is not one time when the throughput reaches zero. The stability and short-term fairness performance is also satisfying.

Because there is some collision at node 2, the aggregate throughput is 496.682 Kbps which is less than that

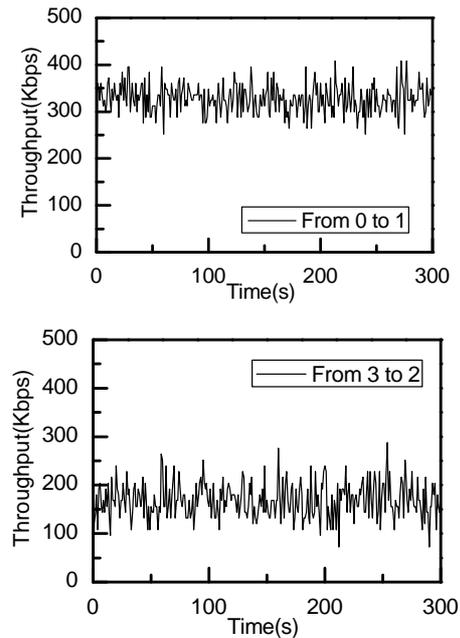


Figure 5. Throughput of two TCP flows in the CDMB-MAC network, $d_1 = 199$ m, $d_2 = 155$ m, $d_3 = 200$ m.

in that in **Figure 3**. And since there is not collision at node 1, throughput of session 1 is about two times that of session 2. As mentioned above, the node mobility can also improve the network performance. Here if the node 3 move only one meter towards node 2, we can also achieve an ideal result similar to that shown in **Figure 3**.

4) Case 4: $d1 = d2 = d3 = 150m$: Different from the scenarios above, all nodes are within the carrier sense range of one another in this scenario. **Figure 8** shows the throughput of the two flows in the CDMB-MAC-based network. Here, the average throughput of the two flows is 354.881 and 359.201 Kbps, respectively. In the 300 s lifetime of the two flows, there is not one time when the throughput reaches zero. More important, we are satisfied with the stability and short-term fairness performance.

Note that node 3 can decode the CTS frame from 1 in this scenario. Therefore, if node 1 is receiving the TCP-Data from 0, node 3 does not transmit an RTS frame. This is also true for node 0. In other words, if one flow has finished the RTS/CTS handshake successfully, it can communication without collision. However, a large value of RRC and fixed size of backoff window will give a high opportunity to two flows to access the channel.

5.2. Adaptivity Analysis of CDMB-MAC

According to simulation results above, we can find if $CINR$ of the receiver is greater than $CPTresh$ or correlative channel contenders are within the carrier sense range of one another, that is

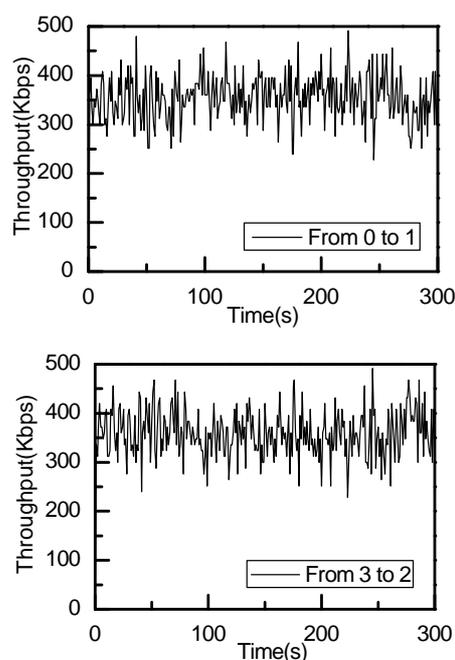


Figure 6. Throughput of two TCP flows in the CDMB-MAC network, $d1 = d2 = d3 = 150 m$.

$$\begin{cases} d1 + d2 > 1.77828d3 \\ d2 + d3 > 1.77828d1 \end{cases} \quad (4)$$

or

$$d1 + d2 + d3 < 550m \quad (5)$$

CDMB-MAC can provide perfect long-term and short-term fairness and stability performance while maintaining a good aggregate. Especially, from Equation (4) and (5), we can deduce a following important fact. In spite of the value of $d2$, if $d1 < 198m$ and $d3 \leq 198m$, we can acquire ideal TCP performance. In addition, if the value of $CINR$ is greater than $CPTresh$ at a MAC receiver, and its contending nodes do not meet this requirement, we can also acquire good TCP performance as the result presented in Case 3. And the TCP parameter, known as maximum window size (*window*), only has a small effect on the performance. We argue that these conditions are not the defect of our CDMB-MAC scheme. On the contrary, they actually provide a deployment guide for ad hoc networks. If the network topology does not satisfy the requirement of Equation (4) or (5), we can utilize node mobility or power control technology to improve network performance, which need further investigation.

6. Conclusions

In this article we focus on the following question: How to acquire fair and stable TCP communication in multi-hop ad hoc networks? Based on the collision detection mechanism of IEEE 802.11, this article has proposed a new CDMB-MAC scheme based on IEEE 802.11. CDMB-MAC replaces the BEB algorithm with a p -persistent transmission strategy and adopts a lot of retry attempts when a node cannot transmit an RTS frame successfully. We have analyzed CDMB-MAC performance from fairness, stability and efficiency points of view in some classic scenarios that are known to lead to fairness issues. Simulation results show if at the MAC layer $CINR$ of the receiver is greater than $CPTresh$ or correlative channel contenders are within the carrier sense range of one another, CDMB-MAC can provide perfect long-term and short-term fairness and stability performance while maintaining a good aggregate.

More importantly, we provide a guide of node movement and deployment. Since the ad hoc network is a self-organizing network, nodes especially the MAC layer's receiver should not move completely blindly and arbitrarily. In the future, we will design a movement algorithm according to this guide and an adaptive algorithm to adjust the transmission probability to acquire optimal performance in different topologies.

REFERENCES

- [1] IEEE STD. 802.11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 1999.
- [2] S. G. Xu and T. Saadawi, "Does the IEEE 802.11 MAC Protocol Work Well in Multi-hop Wireless ad Hoc Networks?" *IEEE Communications Magazine*, Vol. 39, No. 6, 2001, pp. 130-137. [doi:10.1109/35.925681](https://doi.org/10.1109/35.925681)
- [3] S. G. Xu and T. Saadawi, "Revealing the Problems with 802.11 Medium Access Control Protocol in Multi-hop Wireless ad Hoc Networks," *Computer Networks*, Vol. 38, 2002, pp. 531-548. [doi:10.1016/S1389-1286\(01\)00273-0](https://doi.org/10.1016/S1389-1286(01)00273-0)
- [4] J. He and H. K. Pung, "Fairness of Medium Access Control Protocols for Multi-hop Ad Hoc Wireless Networks," *Computer Networks*, No. 48, 2005, pp. 867-890. [doi:10.1016/j.comnet.2004.11.020](https://doi.org/10.1016/j.comnet.2004.11.020)
- [5] C. Chaudet, G. Chelius, H. Meunier and D. Simplot-Ryl, "Adaptive Probabilistic NAV to Increase Fairness in Ad Hoc 802.11 MAC layer," *Proc. of the MedHoc NET*, 2005.
- [6] T. Razafindralambo and I. Guérin-Lassous, "Increasing Fairness and Efficiency Using the MadMac Protocol in Ad Hoc Networks," *Ad Hoc Networks*, No. 6, 2008, pp. 408-423. [doi:10.1016/j.adhoc.2007.03.003](https://doi.org/10.1016/j.adhoc.2007.03.003)
- [7] Network Simulator 2 (ns2), <http://www.isi.edu/nsnam/ns>.
- [8] D. B. Johnson, "Routing in Ad Hoc Networks of Mobile Hosts," In *Proc. of the IEEE Workshop on Mobile Computing Systems and Applications*, 1994, pp. 158-163.
- [9] R. Jain, A. Duresi and G. Babic, "Throughput Fairness Index: An Explanation," *ATM Forum/99-0045*, Feb. 1999, <http://www.netlab.ohio-state.edu/~jain/atmforum.htm>.
- [10] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu and J. Jetcheva, "A Performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols," In *Proc. of Mobile Computing and Networking*, 1998, pp. 85-97.