International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-2 Number-2 Issue-4 June-2012

Congestions control through cloud computing with MANET

Ajey Singh[1], Maneesh Shrivastava[2]
Department of Information Technology[1,2]
Lakshmi Narain College of Technology Bhopal, India[1,2]

Abstract

Adhoc network is collection of temporary nodes that are capable of dynamic forming temporary network, self-organize, and infrastructure less with nodes contains routing capability. As cloud computing services rapidly expand their customer base, it has become important to share cloud resources, so as to provide them economically. In cloud computing services, multiple types of resources, such as processing ability, bandwidth and storage, need to be allocated simultaneously. If there is a surge of requests, a competition will arise between these requests for the use of cloud resources. This leads to the disruption of the service and it is necessary to consider a measure to avoid or relieve congestion of cloud computing environments. This dissertation proposes a new congestion control method for cloud computing environments which reduces the size of packet drop rate for congested resource type instead of restricting all service requests as in the existing networks. We apply Queue based technique and differentiate between the queue based technique with cloud queue based technique .Here we use NS-2.31 simulator for simulation of MANET and take comparative analysis between cloud and wireless network mechanism.

Keywords

Mobile adhoc network, Cloud Computing, QBT

1. Introduction

An ad hoc network is a network with completely self-organizing and self-configuring capabilities, requiring no existing network infrastructure or administration Due to the instability and shared wireless channels, ad hoc networks may suffer from impairments, e.g., route failures, drops due to medium Access Control (MAC) contentions and interference, and random channel bit errors. Route failures, which may significantly affect the network performance, have been considered an important research issue for a long time. We have also investigated this problem, which includes an optimization of the TCP protocol, considering route failures in dynamic ad hoc networks, W. B. Zhu and X. M. Zhang et al. [1]. Multi hop ad hoc networks that adopt IEEE Standard 802.11 MAC layer suffer from contention and the hidden-node problem.

A contention problem in wireless networks occurs when multiple adjacent nodes contend for a shared channel to transmit their packets. Another problem is the hidden-node terminal problem. When two nodes communicate with each other, other nodes that are within the interference area of these two nodes cannot access the channel. As a result, when a node attempts to transmit a packet, it should contend with the neighbour nodes that are not adjacent to access the wireless channel. The extended hidden-terminal problem, J. Li, C. Blake, D. D. Couto et al. [2] is a representative issue that results from the aforementioned property. In this problem, some node may not reply to request-to-send (RTS) packets from a neighbour node, because it cannot access the channel, which has been occupied for communication with some other neighbour. If an RTS sender cannot receive a clear-to-send (CTS) reply within a maximum number of retransmissions (seven times in general), it regards this situation as a link failure, and drops the data packets.

1.1 MANET background information

In this section we will briefly review some important concepts in MANET, which will allow us to better understand congestion in MANET and are mainly used in simulations.

1.1.1. Carrier Sensing Multiple Access (CSMA)

In wireless networks, a community of nodes share a single transmission medium. To avoid collision and better utilize the bandwidth, some kind of medium access control (MAC) protocol is needed. Carrier sensing multiple access (CSMA) is a random access protocol, which allows users to transmit data in a none predetermined way.

CSMA schemes require a user to be sure the medium is idle before the transmission. This is called carrier sensing. If the medium is busy, the user has to back-off for a random period and then re-sense. The

random period is to minimize collision since other users may also want to take the medium at the same time. Once the channel is idle, the user can start transmission.

In mobile ad-hoc networks, CSMA schemes are practically used, for example, IEEE 802.11, Bianchi et al. [3] and SMAC, W. Ye et al. [4]. We will discuss a little about IEEE 802.11 in the following content.
The distributed coordination function (DCF) of IEEE 802.11 is essentially a carrier sensing multiple access with collision avoidance (CSMA/CA) scheme. In addition to physical sensing, it also employs a technique called virtual carrier-sensing. Virtual sensing is realized by a pair of control frames request-to-send (RTS) and clear-to-send (CTS).

Figure 1-1 illustrates the mechanism of RTS/CTS. Node S 1 sends a RTS frame to node S 2 before the real data transmission. Node S 0 also receives the RTS and is blocked by it. Upon receiving the RTS, node S 2 broadcasts the CTS frame to its neighbours. Thus, node S 3 is also blocked. Node S 1 starts transmitting data once receiving the CTS frame from node S 2.

The RTS/CTS mechanism is to deal with hidden terminal problems. In Figure 1.1, node S 5 is a hidden node of the transmission from S 1 to S 2, since S 5 is beyond the interference range of S 2 (two hops). Node S 5 cannot sense the data flow from S 1 to S 2 and will think the medium is idle. If there is no RTS/CTS, node S 5 will directly start sending data packets to S 4. In this case, the ACK frames from node S 4 will be very likely to collide with the data received by S 2. With the use of RTS/CTS, node S 5 won't get the CTS from S4 and cause interference to S 1 and S 2 since S 4 can detect the flow between S 1 and S 2.

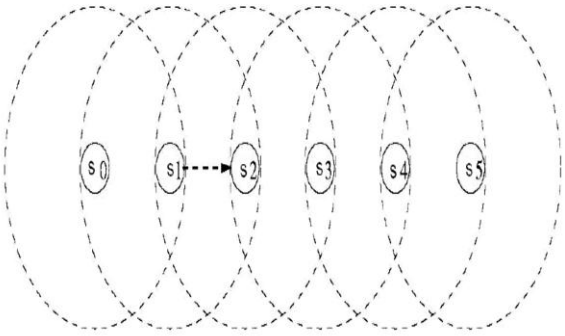


Figure 1-1: RTS/CTS Mechanism

The mechanism RTS/CTS introduces a lot of overhead especially when the data load is relatively low. Thus, sometimes, RTS/CTS is suggested to be disabled when IEEE 802.11 or its variant is used in sensor networks.

1.1.2. Multi-hop Wireless (CSMA Type) Network Capacity

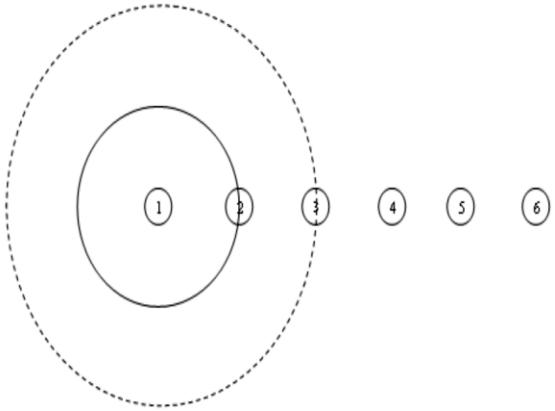


Figure 1- 2: Multi-hop Wireless Network Capacities

Multi-hop wireless capacity is relevant for sensor networks as these networks show multi-hop behavior. Multi-hop capacity alone can create congestion. For example in Figure 1-2 suppose node1 is the source and node 6 is the sink of such a network. Assuming that the nodes that are node neighbors do not interfere with each other, when node 1 transmits, only node 4 and beyond can transmit simultaneously, because it is clear that when node 1 transmits, node 2 cannot transmit. Also node3 is blocked from node's 2 CTS, and if node 4 is transmitting then nodes 5, 6 are blocked by node 4. So an ideal MAC protocol could give a chain utilization of 1/3, Jinyang Li et al. [5] . This is giving a:
C=1/3x (packetsize)/(packetsize+RTSsize+CTSsize +ACKsize)xChannelCapacity
Which is about 0.425Mbps for a Channel Capacity of 2Mbps, and 1500,40,39,47 packet sizes for data packet, RTS,CTS and ACK packets respectively. In case that interference range is up to 2 hops away, the capacity is even worst. In such a case in a 7 node chain node 3 experiences interference from 5 other nodes, while node 1 from only 3 nodes. This means that node 1 has better service rate than node 3 and can inject more traffic to the network that node 3 can handle. In such a case congestion occurs.

2. Proposed Solution and Algorithm

In our approach we control congestion through collision avoidance, we use data link layer channel access technique CSMA/CA, 802.11 and TDMA mechanism, in CSMA/CA technique from a network point of view, one of the primary reasons for using the MCCN mechanism is to avoid network congestion resulting from frequent packet collisions. And also use contention window scheme in CSMA/CA case in our simulation we set minimum contention period and maximum contention period and avoid collision, but 802.11 case we can't using MCCN mechanism so that no any request to send and clear to send information transmitted to all neighbour, so maximum chance to collision comes in our network.

According to problem statement we also apply routing protocol as AODV (ad-hoc on demand distance) routing protocol and for the TCP best performance case we use TCP New Reno mechanism, TCP-Newreno has advantage of its strategy to detect and handle multiple packet loss thereby avoiding continuous retransmission timeouts. It enters into fast-retransmit phase when it receives multiple duplicate packets but it does not exit this phase until all the packets which were out standing at the time it entered fast recovery get acknowledged, Prasanthi. S et al. [6].

Algorithm for our Mechanism
Algorithm for Contention period, Jam Period and data Send Method
// The MAC calls this Channel contention() to enter contention period

```
Channel::contention(Packet* p, Handler* h)
        //Contention Definition
         {
Step 1:  Scheduler& s = Scheduler::instance();
//Create Instance variable s
Step2:          double now = s.clock();
                 // through S we get System Time
Step3:          if (now > cwstop)
                // Check Congestion Window
                {
                        //If True
                        cwstop  = now + delay;

                        numtx  = 0;
                // initialize number of Tx =0
                }
Step4:                  numtx++;
                // Increase no. of Tx Value
        }
```

Jam the channel for a period txtime

```
Channel::jam(double txtime)
                // Jam Method Definition
{
        // without collision, return 0
Step1:  double now = System.clock();
// Now Gives Current System Time
Step2:    if (txstop  > now)
            // Check Condition
          {
            // If True
          txstop = max(txstop, now + txtime);
// new Tx Stop is Max of New Value
          return 1;
// Error Less
                        }
Step3:  txstop = now + txtime;          // Else
normal Tx Stop time
Step4:          return (now < cwstop);
         }
```

Data send through Send method

```
Step1: Channel::send(Packet* p, double txtime)
        // Send method Definition
        {
Step2:   double drop ;
        //Variable declaration Drop
Step3:  double now = clock();
                // Call System Time

        // busy = time when the channel are still
busy with earlier tx
Step4:  double busy = max(txstop, cwstop);

Step5:    if (now < busy) {
                //Check Condition
        // if still transmit earlier packet, pkt, then
corrupt it
Step6:          if (pkt->time > now) {
                access(pkt)->error()            |=
EF_COLLISION;
Step7:                  if (drop) {
                // pass drop value
                                cancel(pkt);
                //pass to cancel packet
                                pkt = 0;
        // initialize pkt
                        }
                }
```

```
Step8:          if (drop) {
        //If Drop Than return True
                        return 1;
                }
        }
Step9:          pkt = p;                        //p pass to pkt value
                trace ? trace->recv(p, 0) : recv(p, 0);
                return 0;
 }
```

3. Simulation Environment

The simulator we have used to simulate the ad-hoc routing protocols in is the Network Simulator 2 (ns) from Berkeley. To simulate the mobile wireless radio environment we have used a mobility extension to ns that is developed by the CMU Monarch project at Carnegie Mellon University.

4. Network Simulator

Network simulator 2 is the result of an on-going effort of research and development that is administrated by researchers at Berkeley. It is a discrete event simulator targeted at networking research. It provides substantial support for simulation of TCP, routing, and multicast protocols.

The simulator is written in C++ and a script language called OTcl2. Ns uses an Otcl interpreter towards the user. This means that the user writes an OTcl script that defines the network (number of nodes, links), the traffic in the network (sources, destinations, type of traffic) and which protocols it will use. This script is then used by ns during the simulations. The result of the simulations is an output trace file that can be used to do data processing (calculate delay, throughput etc) and to visualize the simulation with a program called NAM is a very good visualization tool that visualizes the packets as they propagate through the network.

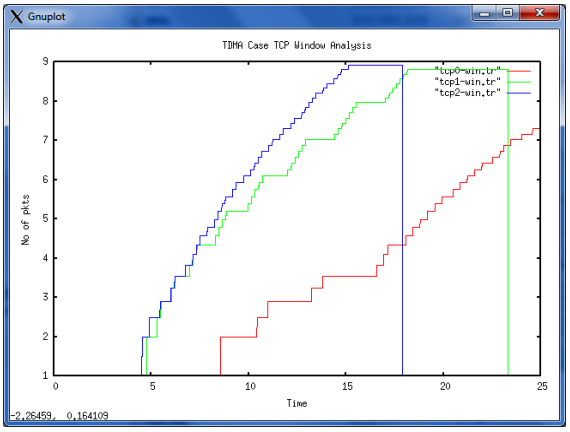
5. Result

We get Simulator Parameter like Number of nodes, Dimension, Routing protocol, traffic etc. According to below table 5.1 we simulate our network.

Table 5.1 Simulation parameter

Number of nodes	**30**
Dimension of simulated area	800×600
Routing Protocol	AODV
Simulation time (seconds)	25
Mac Layer property	802.11 , TDMA,CSMA/CA

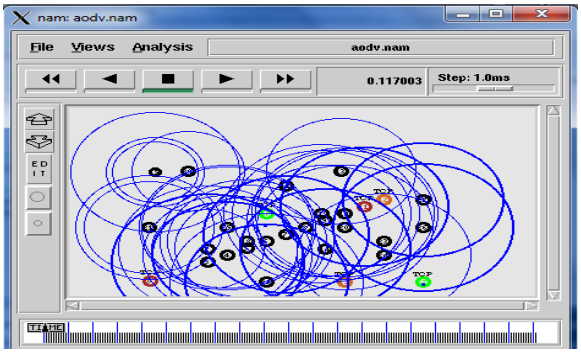
Here we show figure 5.3, the result of TCP (Transfer Control Protocol) packet flow analysis, in our simulation we take 30 mobile node with MAC as TDMA with three TCP connection, time division access case all the sender node send data according timely manner that case no any collision occurs on the network but heavy traffic case end to end delay has increases, in that diagram tcp0 , tcp1 and tcp3 packet transmitted through the genuine sender to intended receivers, and according to resultant graph our simulation maximum time is 25 sec. graph shows all tcp data start sends nearby 5th sec. tcp2 and tcp1 maximum data send in time within time 10th sec. to 18th sec. but tcp0 flow start at the time nearby 8th sec. ant maximum data send at the time of 20 to 25th sec. that case data send in timely manner so our flow start different time units.



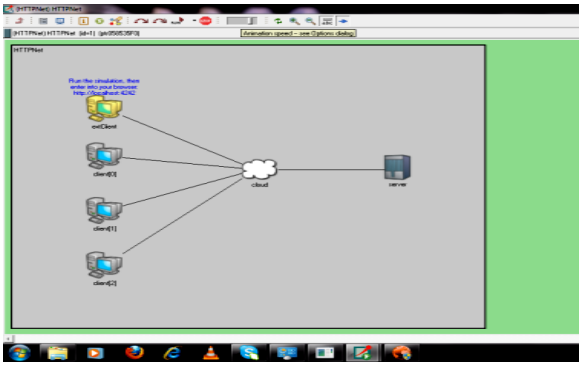
5.1: Nam visualization

The simulation described in this project was tested using the ns-2 test-bed that allows users to create dynamic topologies. By changing the logical topology of the network, ns-2 users can conduct tests in an ad hoc network without having to physically move the nodes.

In our simulation we use thirty mobile nodes with random deployment and random motion of each node, here we create three TCP senders and two UDP sender nodes with 10 connections, in this figure blue circle shows radio range of the particular node. If routing packet broadcast via the sender node firstly check the neighbour node belongs to radio range or not if neighbour node is in radio range so our routing packet send to neighbour else not. After the routing discovery process actual FTP data transmit through shortest path

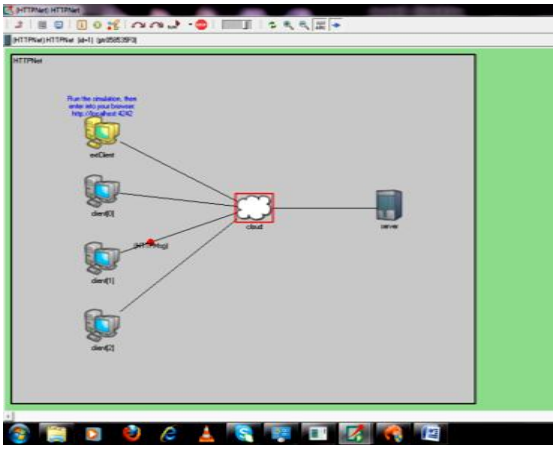


5.2: Result1



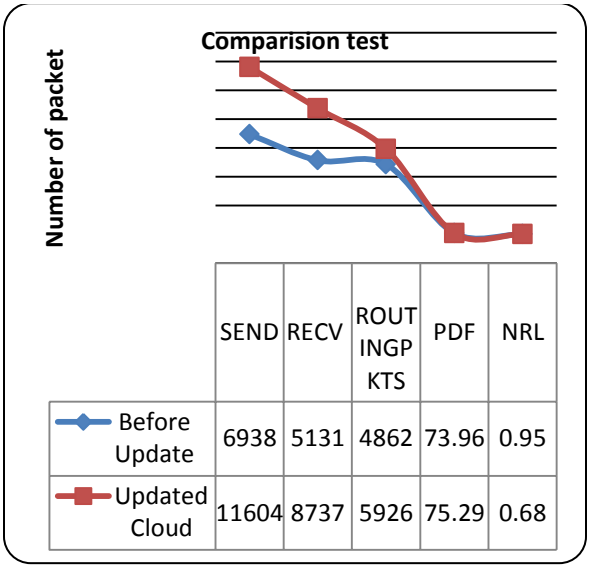
5.3: Result2

5.4: Result3



5.5: Result4

By using cloud with queue management technique we find that packet drop rate is reduced.



	SEND	RECV	ROUTINGPKTS	PDF	NRL
Before Update	6938	5131	4862	73.96	0.95
Updated Cloud	11604	8737	5926	75.29	0.68

5.6: Result5

Table 5.2: Overall Summaries

Overall Summary

Parameter		Before Update	Updated Cloud
SEND	=	6938	11604
RECV	=	5131	8737
ROUTINGPKTS	=	4862	5926
PDF	=	73.96	75.29
NRL	=	0.95	0.68

References

[1] W. B. Zhu, X. M. Zhang, and N. N. Li, "Improve TCP performance with link-aware warning method in mobile ad hoc networks," in Proc. IEEE WICOM, 2008, pp. 1–4.

[2] J. Li, C. Blake, D. D. Couto, H. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in Proc. ACM MobiCom, Jul. 2001, pp. 61–69.

[3] Bianchi,G. Dipt. di Ingegneria Elettrica, Palermo Univ, " Performance analysis of the IEEE 802.11 distributed coordination function", IEEE Journal ,2000.

[4] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks" in Proceedings of IEEE INFOCOM'02, June 2002.

[5] Jinyang Li, Charles Blake, Douglas S. J. De Couto, Hu Imm Lee and Robert Morris, "Capacity of Ad Hoc Wireless Networks", Mobicom'01 2001.

[6] Woo, Alec, and David E. Culler. "A transmission control scheme for media access in sensor networks." In Proceedings of the 7th annual international conference on Mobile computing and networking, pp. 221-235. ACM, 2001.

Ajey Singh completed his B.Tech (I.T.) from university of chitrakoot, satna. He enrolled M.Tech (I.T.) in LNCT, Bhopal. His research area in Security in Cloud computing.