Extending the OPNET Simulation Environment for Mobile Agents in Network Congestion Control

Nguyen Hong Van[1], Oliver Popov[2]
[1]DSV, Stockholm University and KTH, Forum 100
SE-164 40 Kista, Sweden
[2]ITM, Mid Sweden University, Holmgatan 10,
SE-851 70, Sundsvall, Sweden

si-hvan@dsv.su.se (Nguyen Hong Van)

Abstract

Despite the continuous developments concerning the performance of the global Internet, and hence networking in general, congestion management remains to be a significant challenge. This is due to the heterogeneity of (1) the infrastructure (wired and wireless), (2) network flows (responsive and unresponsive to congestion), and the enormous growth of users and the variety of services. There are number of solutions proposed to tackle congestion, especially concerning the co-existence of responsive and unresponsive transport protocols. One of those, termed as Combined Model for Congestion Control (CM4CC), in addition to the set of classical congestion control mechanisms employed by TCP, introduced mobile agents to manage the unresponsive flows, such as UDP.

The paper goes beyond the theoretical foundations of CM4CC, established in a few early articles, by using the simulation paradigm to validate the model. In order to do so, various scenarios are implemented in the simulation environment provided by the Optimized Network Engineering Tool (OPNET). The results of the simulation study clearly prove that CM4CC, which is a collaborative effort by TCP feedback mechanisms and mobile agents monitoring and control of the network and in particular the behaviour of UDP sources, is more than a promising venue towards a comprehensive congestion management. Inter alia, the focus is on the extensions made to the OPNET to accommodate mobile agents beyond the multi-tier system, and thus extending the original simulation capacity to the areas such as congestion management, network performance enhancement and stability.

Keywords: Responsive and unresponsive flows, Congestion control, Mobile agent packet, Simulation.

Presenting author's biography

Nguyen Hong Van. Ms. Van is a Ph.D. student at the Department of Computer and Systems Sciences, Stockholm University and Royal Institute of Technology. She works for the Ministry of Science and Technology of Vietnam, and her academic stay in Sweden is made possible through a scholarship from the Swedish government, which is administered by SIDA.

ISBN 978-3-901608-32-2

1 Introduction

TCP and UDP protocols still dominate the transport layer in the IP networks and on the Internet. While the former is a complex, closed-loop protocol used mainly when there is a need for reliable transmission, the later, which is light-weight, unreliable and open-loop, deals with multimedia applications characterized by the large volumes of data and interactivity. Out of the two protocols, TCP is still the major transport vehicle on the Internet, with about 80 % of the flows. There are more then two hundred modifications of the original protocol trying to absorb the immense growth in users and services, and changes in the infrastructure (from copper to fiber, and from wired to wireless).

On the other hand, there have been hardly any attempts to modify UDP. Understandably so, considering the applications using the protocol and the objectives relative to their optimal performance - high speed and low delays.

One of the key problems on the Internet and in networking is to make TCP and UDP work together, bearing in mind that the first one is “socially or network responsible”, while the second one is irresponsible, viz. aggressive and greedy. Whenever TCP and UDP compete for network resources, the later wins that usually makes networks underperform, unfair, and unstable.

The responsive behavior of TCP to address the problem of congestion management has been studied and subsequently bettered based on two different strategies. One is to keep the E2E semantics, or so called host-centric method, where alterations are made in the protocol stacks at the end systems. The other one, denoted as router centric, attempts to reduce the contention for network resources through changes in the architecture and the organization of the routing devices, especially buffers, both in a passive and active way [4]. There are also recent efforts to combine some of the distinct features of both protocols, for example congestion control from TCP and unreliability from UDP, in a protocol designated as Datagram Congestion Control Protocol (DCCP) [11] or to isolate TCP and TCP-friendly flows in tunnels, so they can be protected from irresponsive flows [4]. Another scheme such as BLACK [7] employs blacklisting of the unresponsible flows, where the packet drop for high-bandwidth flows is proportional to the deviation from the fair share rate.

It is not the intention of the article to enumerate all of the endeavors proposed so far to resolve the problems with respect to congestion management. This has already been done in numerous articles [2, 3, and 4]. Hence, we shall point out the main characteristics of our approach represented by the Combined Model for Congestion Control (CM4CC) [2], and then proceed with the elaboration of the main objectives (1) to use simulation in proving the validity of the model, and (2) to extend the basic simulation environment, if necessary, so it may accommodate all of the strategies and mechanisms used in CM4CC.

2 Overview of the Combined Model for Congestion Control (CM4CC)

CM4CC builds on the host-centric standard, thus preserving the E2E semantics and the whole family of TCP algorithms (slow start, congestion avoidance, and AIMD) for congestion control [2]. It simply acknowledges that TCP and TCP-friendly that are behind responsive flows. To make these flows work with the unresponsive or UDP flows, we reach for the mobile agent paradigm. The relevance of mobile agents in network management, e-commerce systems, telecommunication, and information collection, storage and retrieval has been widely studied, examined and advocated for [1].

In CM4CC mobile agents exhibit different functionalities such as network monitoring, information collection on the network status and hence congestion, assessing the state of the network and finally taking whenever necessary appropriate actions to remedy the problems in the network related to contention for network resources among different, both responsive and unresponsive flows. In fact, mobile agents play a major role in dealing with the unresponsive (UDP) flows.

The fundamental attributes and functionalities of CM4CC were explained in [2] primarily for wired networks, and then extended for wired ones. Since all the pertinent details and operations of CM4CC were presented and explicated in [2, 12], only to the most important features are briefly outlined here.

Usually, a packet loss is one of the indicators of congestion. In case of TCP, this means a back-off to a lower rate of packet infusion into the network (either drastic to system restart or moderate to one half of the current congestion window). However, these measures do not apply to unresponsive flows that should be also tamed down to defuse the state of congestion. The situation, as CM4CC affirms calls for mobile agents who move across the network to collect and process information related to unresponsive flows. Based on the information from the network, which is in a way a synthetic substitute for the natural feedback mechanisms in TCP, the sending rates of the unresponsive flows will be lowered (provided there is a congestion) via a traffic shaper at each source host.

Out of the seven mobile agents used in CM4CC, three deal with congestion avoidance and are designated as monitor agent (MoA), management agent (ManA), and control agent (CtrlA). ManA coordinates activities that include policies for unresponsive flows, loss and maximum allowed sending rates, and creation of control agents when necessary. MoA detects active unresponsive flows and gathers information about sending and receiving rates of non-TCP flows. Control agents (CtrlAs) have responsibility to (1) move to source hosts where sending rates of non-TCP flows exceed maximum allowed sending rates; and (2) control and adjust indirectly sending rates of non-TCP flows in an indirect manner through traffic shapers placed at the corresponding source hosts.

3 CM4CC in OPNET Environment

OPNET [8] is a network simulation tool used rather widely in commercial and research communities. Any further reference to the OPNET in the paper, addresses the entire simulation environment inclduing the OPNET modeler.

While OPNET is a reach simulation tool, many issues related to mobile agents are not yet supported by built-in utilities and objects in the standard libraries. For instance, the use of mobile agent in network management is based on a utility named *multi-tier application* [9, 10]. The utility takes care of network performance metrics such as delay, response time, and throughput [8]. However, this is not sufficient to model and assess the complex behavior of mobile agents, and therefore not quite suitable to describe and explore the full potential of the CM4CC model.

In essence, a mobile agent is a specific piece of programming code that is able to move across the network. The state and the associated data, carried by a mobile agent, need to be encapsulated into a packet in a serialized form.

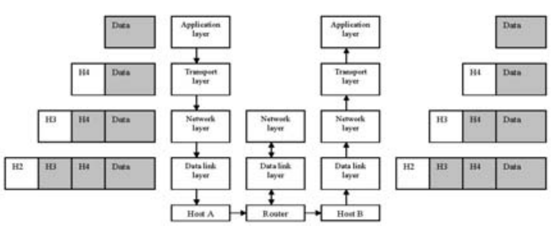

Fig. 1 TCP/IP stack in IP networks

There are some differences between the TCP/IP stack implemented in a real IP networks (Figure 1), and the corresponding TCP/IP stack in OPNET (Figure 2). The differences are mainly reflected in the process of packet composition (encapsulation) and decomposition (decapsulation)

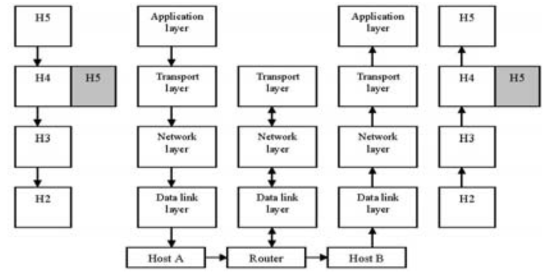

Fig. 2 TCP/IP stack in OPNET

In the case of OPNET, applications do not generate real data packets, which imply that the built-in utilities and objects in OPNET standard libraries do not maintain and store complete information relevant to the packet that moves across the network. In summary, these utilities and objects do not suffice to stipulate and stimulate the entire spectrum of mobile agent activities in CM4CC. An instance of this limited ability is the failure to induce the behavior of a monitoring agent that carries the information on sending and receiving rate.

There are few problems to be addressed and resolved in OPNET before we proceed to simulate the operation of CM4CC. These questions are the representation of a mobile agent in OPNET and the inducement of agent's mobility (by using the complete list of network nodes that a mobile agent has to visit in network, i.e. how mobile agent can move to the sequential list of hosts in OPNET); storage and preservation of data collected by an agent from the network during his travel and mobile agents co-operation and co-ordination when they exercise congestion control.

4 Extending OPNET for CM4CC

The simulation of mobile agents requires a packet in a special form, which for the sake of the argument will be termed as mobile agent packet (MA packet). In fact, without loss of generality one may consider that the concepts of a mobile agent and a mobile agent packet are identical. The header of the MA packet should contain information on the packet itinerary (list of all nodes to be visited during a network trip including the next intermediate host. One should also make all the provisions to avoid revocation of the packet at any host until the whole network or part of it is traversed as necessary. Each MA packet needs to store the data gathered during mobile agent or packet travel across the network.

The structure of the MA packet is

Type	Dest_addr	Dest_ subnet	Visit_addr _list	SR	RR	MA SR

where

- *Type* is a constant 32 bits long that stands for the type of a mobile agent.
- *Dest_addr* and *dest_subnet* are variables 32 bits long that store the address of intermediate next host that mobile agent will visit.
- *Visit_addr_list* is a variable, 32 bits long, that contains the list of the remaining host addresses that mobile agent will visit.
- *SR, RR* and *MASR* are aggregated data arrays, which store sending, receiving, and the maximum allowed sending rates for UDP flows respectively. Each element of the array is a variable 32 bits long

The behavior of mobile agents during the congestion avoidance phase is defined by CM4CC. Their comprehensive description provided by MA packets is given on Figure 3. In the beginning, the Monitoring Agent (MoA) is created in form of MA packet by the packet generator integrated in the home host, which is actually the place where all mobile agents are created and eventually terminated upon their return.

The information related to the address of the first host visited and is placed at the dest_addr and dest_subnet fields. The information on the rest of the list (a sequence of hosts that need to be visited) is in the visit_addr_list field. Each host visited provides the mobile agent MoA with info on UDP sending and receiving rates that either originate or terminate at the specific host and are saved in the arrays SR and RR.

The migration of mobile agent to a new host causes the variables dest_addr and dest_subnet to be set to the values from the visit_addr_list, which also decreases the list of elements for one. MoA returns home (Home Host) when the list of all host is exhausted, reports the data to ManA and waits for the next assignment or data collection.

Based on the sending and receiving rates of UDP flows provided by MoA, ManA calculates packet loss rate. If the loss rate is greater than the Allowed Loss Rate (ALR), ManA will calculate the Maximum Allowed Sending Rate (MASR) and place it in MASR. If there exists at least one UDP flow whose sending rate has to be regulated, ManA creates new control agent (CtrlA) that follows almost the same sequence of actions as MoA. The CtrlA is terminated once it returns to the home host.

When UDP source host receives information on MASR from CtrlA, it informs the traffic shaper. With the absence of traffic shaper in OPNET, an object had to be built and integrated in the UDP host. The traffic shaper used in our simulation study utilizes a leaky bucket mechanism. It is worth mentioning that just like the UDP source and destination host objects, all other entities used in the simulation such as home hosts, links, switches and routers are designed and constructed specifically for the validation of the CM4CC.

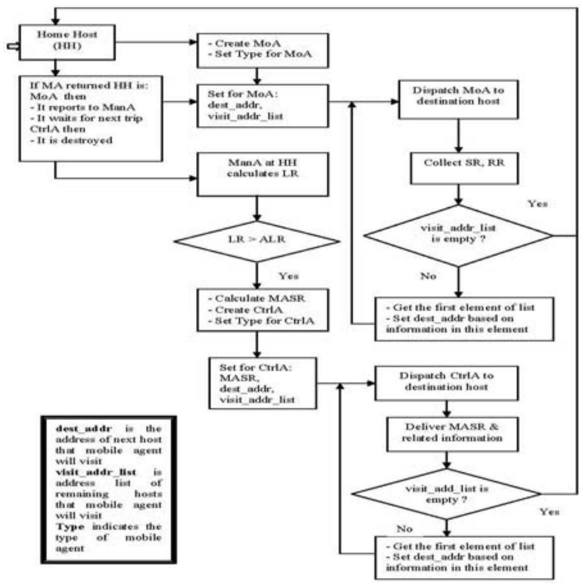

Fig. 3 Mobile Agent based Congestion Control Protocol (MACCP)

For instance, routers used in the simulation route packets from the input into the output stream and drop packets when the queue is full. UDP source hosts have the capability to transmit data to UDP destination hosts and vice versa, to send and receive mobile agents, and has an integrated traffic shaper and a packet generator.

Home hosts are capable of creating different types of mobile agents, launch them into network at certain intervals, or if required, terminate agents when they are done and back home.

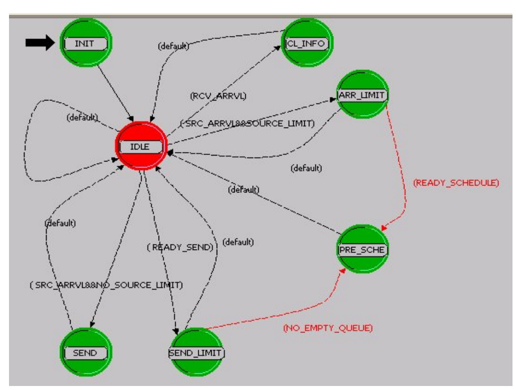

Fig. 4 The Process Model

5 Simulation Experiments

Figure 5 shows scenario 1, with one UDP and one TCP flow. MA_home is the home host of the mobile agents, UDP1_Send is the source host of the UDP1 flow, and UDP1_Receive is the corresponding destination host. TCP_Send and TCP_Receive are the source host and the destination host of the TCP flow respectively. The link capacity between the Router1 and the Router2 is 2 Mbps, while the link capacity between Router2 and UDP1_Receive is 1 Mbps. The capacity of the other links is 10 Mbps each.

In scenario 1, UDP1 is the unresponsive flow and is generated at the constant bit rate 2.5 Mbps for the whole duration of the simulation run. TCP (Newreno variant) flow is generated by an application that infuses data into the network during the entire simulation interval.

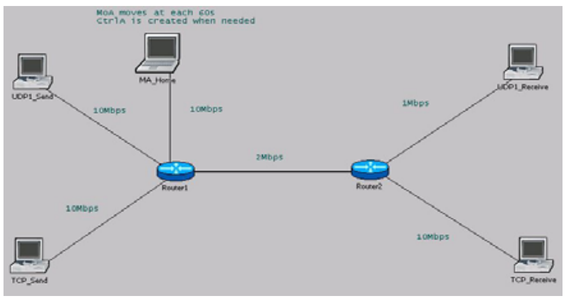

Fig. 5 Simulation scenario 1

MoA is generated periodically each 60s after the start of the simulation. The period of reduction is set to 60s, and is defined as the time during which the traffic shaper limits the sending rate of UDP1 flow to the value of MASR as received from the control agent. When the period of reduction is over, the UDP1 sending rate is not any more limited until new MASR is being calculated. The total simulation time is set to 1200s. The UDP1 flow allowed loss rate (ALR) is set to 0.1 (or 10%).

The routers in the simulation model have two FIFO input queues, one for TCP flows and one for UDP flows.

The partial trace of mobile agents activities in Scenario 1 is shown on Figure 6. Lines 2 and 3 depict the information of UDP1 sending rate (SR) and receiving rate (RR) collected at 60.00s and 65.81s. Since the loss rate calculated from the data gathered by MoA is greater than ALR (line 4), CtrlA is created, which delivers MASR (1262626.26 bits/sec bits/sec) to UDP source host at time 65.81s.

In line 9, MoA starts its second trip to collect SR and RR of the UDP flow. While there is some reduction of the sending rate (the loss rate still is higher than ALR), CtrlA has to be created again (line 12) to control the UDP flow sending rate.

1 Travel No : 1 of MoA to collect data
2 MoA collect SR 2525252.53 bits/sec of flow UDP1 at time 60.00
3 MoA collect RR 1010101.01 bits/sec of flow UDP1 at time 65.81

4 CtrlA delivers MASR 1262626.26 bits/sec of flow UDP1 at 65.81
5 Size of UDP packet is 12500 bytes
6 Number of packets is sent in one sec is 12.63
7 Interval between UDP packets is 0.08
8 757 packets are sent within regulated period

9 Travel No : 2 of MoA to collect data
10 MoA collect SR 1275380.06 bits/sec of flow UDP1 at time 120.00
11 MoA collect RR 1010101.01 bits/sec of flow UDP1 at time 122.71

12 CtrlA delivers MASR 1137639.02 bits/sec of flow UDP1 at 122.71
13 Size of UDP packet is 12500 bytes
14 Number of packets is sent in one sec is 11.38
15 Interval between UDP packets is 0.09
16 682 packets are sent within regulated period

17 Travel No : 3 of MoA to collect data
18 MoA collect SR 1149130.32 bits/sec of flow UDP1 at time 180.00
19 MoA collect RR 1010101.01 bits/sec of flow UDP1 at time 180.71
20 CtrlA delivers MASR 1125014.04 bits/sec of flow UDP1 at 180.72
21 Size of UDP packet is 12500 bytes
22 Number of packets is sent in one sec is 11.25
23 Interval between UDP packets is 0.09
24 675 packets is sent within regulated period

Fig. 6 Trace of Mobile Agents in Action

Figure 7 indicates that the throughput of UDP1 is always 1 Mbps (regardless of whether or not mobile agents are used for congestion control).

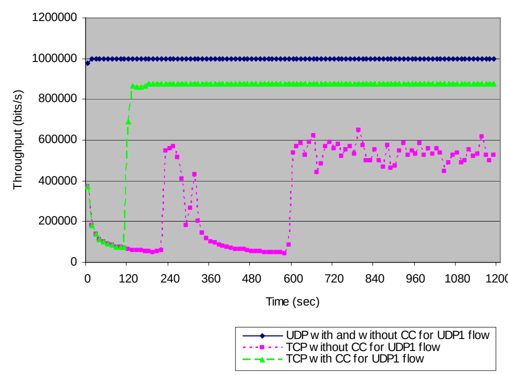

Fig. 7 UDP1 and TCP flows throughput with and without congestion control

One may notice that 1 Mbps is smallest link capacity on the path to UDP1 destination host, which is natural due to the UDP greediness.

Scenario 2 with two UDP flows is presented in Figure 8. In this case, UDP1_Send and UDP2_Send are the source hosts, while UDP1_Receive and UDP2_Receive are the destination hosts of UDP1 and UDP2 flows respectively. Both UDP flows are generated, during the entire simulation interval, at the constant bit rate 2.5 Mbps. The capacity of the link between Router1 and Router2 is 4Mbps, while of the other links is 10 Mbps. The other parameters are same as in Scenario 1.

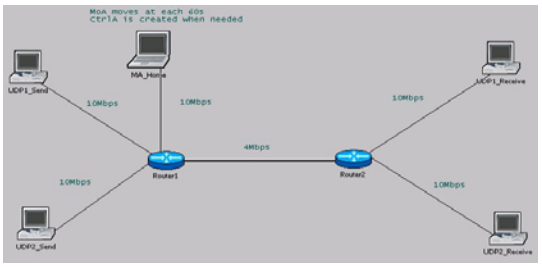

Fig. 9: Simulation scenario 2

Figure 9 shows the throughput of UDP1 and UDP2 flows forced to use UDP congestion control.

One can realize from Figure 10 that the sending rates of UDP1 and UDP2 are always regulated prior to entering the network by traffic shapers (provided that the loss rate is greater than ALR). In some way, this suggests that the receiving rates of the UDP flows approach the sending rates. Nevertheless, there is still a difference between them since the allowed loss rate (ALR) is set to 0.1, quite acceptable level of loss tolerance due to the needs of multimedia applications.

Figure 11 shows the difference between the UDP sending rates and receiving rates in a network without congestion control for UDP. While both flows are generated with the same constant bit rate, after some simulation time, the UDP1 flow has a good throughput, and UDP2 is limited to the rest of the link capacity between router 1 and router 2.

The results unequivocally showed that the congestion control imposed by the mobile agents on the UDP flows, and thus making them more responsible reduced the overall network vulnerability to impaired performance due to congestion and the risk of congestion collapse.

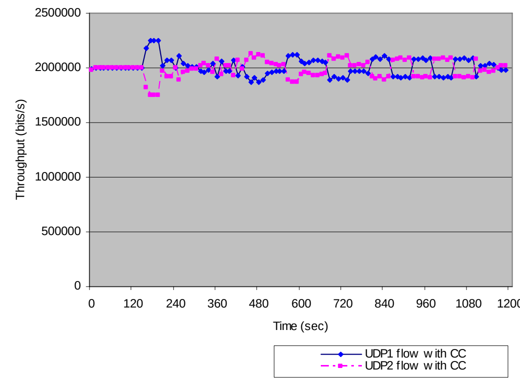

Fig. 9 UDP throughput in presence of UDP congestion control

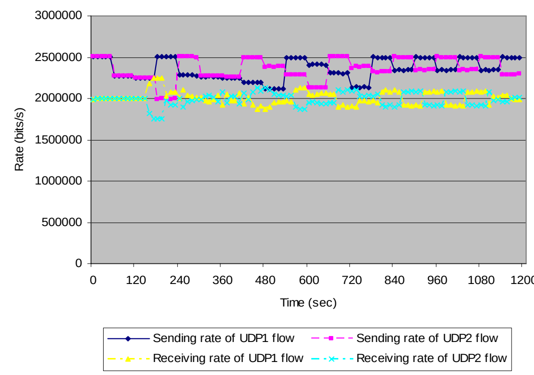

Fig. 10 Sending and receiving rates of UDP flows with congestion control

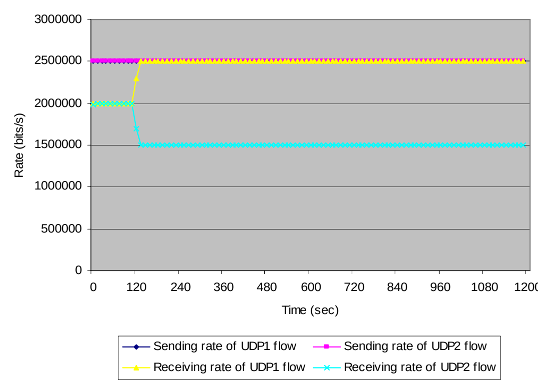

Fig. 11 UDP Sending and receiving rates without UDP congestion control

6 Conclusion

The paper posits several important results concerning the use of mobile agents in solving the problem of congestion management of both, responsive (TCP) flows and unresponsive (UDP) flows. It also makes their co-existence possible by improving the fairness and the stability of the network and hence the global Internet.

In order to prove the feasibility and the validity of the Combined Model for Congestion Management (CM4CC), we used the expressiveness and the power of simulation provided by the OPNET simulation environment.

Nevertheless, the complexity of the mobile agents, and their attributes such as mobility, information collection and analysis, and occasionally the need for a prolonged presence, where not quite suited for the multi-tier application schema available from the OPNET standard library.

Therefore, there was a need to develop our own sent of objects, network entities such as links, routers, switches, and hosts. Even more important was the design of the Mobile agent packet (MA packet), a novel structure that actually may represent a mobile agent at any time and situation on the network. The effort helped us prove that CM4CC works rather well whenever mixed TCP and UDP flows are involved and coexist by making the later more responsive towards the needs of the former and the whole network.

This actually extends the original OPNET tool with the possibilities that enable and facilitate the use of multiagent paradigm concerning (1) CM4CC in our current and future research, and (2) any other network and Internet phenomena.

7 References

[1] D. B. Lange, and O. Mitsuru. Seven good reasons for mobile agents. *Communications of the ACM*, Vol. 42, 3:88-89, 1999.

[2] N. H. Van, O. B. Popov, and I. Popova. Combined Model for Congestion Control -CM4CC. *Journal of Computing and Information Technology (CIT)*, No. 4, 14:337-342, 2006.

[3] R. Pan, B. Prabhakar, and K. Psounis. CHOKe - a stateless active queue management scheme forapproximating fair bandwidth allocation. *INFOCOM 2000 - Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, Vol. 2, 942-951, 2000.

[4] B. Lee, R.K. Balan, L. Jacob, W.K.G. Seah, and A.L. Ananda. Avoiding congestion collapse on the internet using TCP tunnels**.** *Computer Networks: The International Journal of Computer and Telecommunications Networking***,** Vol. 39, Issue 5, June 2002, pp. 207-219

[5] Haitao Wu; Keping Long; Shiduan Cheng; and Jian Ma, Direct congestion control scheme (DCCS) for differentiated services IP networks**.** *GLOBECOM IEEE*, Vol. 4, 2001, pp. 2290 - 2294, 2001.

[6] R. Habib, and B. Bhargava. Network tomography-based unresponsive flow detection and control**.** *Distributed Computing Systems, 2003- Proceedings of The Ninth IEEE Workshop on Future Trends*, May 2003, pp 258 - 264.

[7] G. Chatranon, M.A. Labrador, and S. Banerjee, BLACK: Detection and preferential dropping of high bandwidth unresponsive flows, *Proc. ICC2003*, Vol.1, pp.664–668, May 2003.

[8] http://www.opnet.com.

[9] T. Fahad , S. Yousef , and C. Strange. A Study of the Behaviour of the Mobile Agent in the Network Management Systems *The IEE Networking Conference (PGNet) 2003*, Liverpool, UK, June 2003.

[10] I. Adhicandra, C. Pattinson, and E. Shagouei, Using Mobile Agents to Improve Performance of Network Management Operations, *Postgraduate Networking Conference (PGNET 2003)*, 2003, Liverpool, UK.

[11] E. Kohler, M. Handley, and S. Floyd. Designing DCCP: Congestion Control without Reliability, *Tech. Rep. ICIR*, 2004.

[12] O.B. Popov, and Nguyen Hong Van, Network Congestion Management and the Mobile Agent Paradigm, *Acta Electrotechnica at Informatica,* No.1, Volume 6, 71-75, 2006.

ISBN 978-3-901608-32-2