4

High Performance Computing and Networking (HPCN)

The HPCN group at C-MMACS is engaged in a multi-component programme consisting of algorithm development, code design network and security. It is also responsible for the development, operation and management of computing and networking resources which form the lifeline for research in areas where modelling and simulation play a critical to enabling role.

Highlights

- Denial-of-service Attack Detection through Active Hop-count Mapping
- Random Key Generation Algorithm for Secure Key Distribution
- SGI Origin 3400 Compute Server Upgraded to 24 Processors

Inside

4.1 TCP based Denial-of-Service Attack to Edge Network: an Analysis and Detection

Denial-of-Service (DoS) attacks continue to be a major threat to the secure and reliable operation of the Internet. The focus of our work is on analysis and detection of a special class of DoS attack scenario which can arise as a result of an adversary deliberately violating the end-to-end congestion control algorithms employed in Transmission Control Protocol (TCP). The attack scenario is called as pulse attack and it is mainly targeted to the outbound TCP flows of a network. Unlike most of the widely seen DoS attacks where external machines located at various parts of the Internet are used to generate sustained attack flood, the pulse attack makes use of the transmission power of an internal machine physically located inside the targeted network to generate periodic flood pulses of ON/OFF pattern without compromising the internal machine. The attack is highly illusive to detection as it is practically very difficult to distinguish the ON/OFF flood traffic from highly bursty Internet traffic.

Consider a typical inter-networked setup where an edge-network, like the Local Area Network (LAN) of an organization, is connected to the Internet through an access router of finite buffer size (normally in the order of 50 packets) and an access link of finite bandwidth. Whenever the traffic rate from the LAN to the Internet exceeds the access link bandwidth, packets are buffered at the access router and once the buffer is full, the router drops further incoming packets. Though packet drop can adversely affect the performance of any flow, its impact on TCP flows is much more predominant. This is because, if multiple packets are dropped within a transmission window, the TCP sender is likely to undergo an expensive recovery process which is based on Retransmission Timeout (RTO). In RTO based recovery process, the sender not only reduces its transmission rate to the minimum possible value, one packet per Round Trip Time (RTT), but it also has to wait for a minimum period of at least one second before the first lost packet is retransmitted. It is a well-known fact that this recovery process, if triggered frequently, will severely degrade the overall throughput of any TCP flow. This characteristic of TCP flows is exploited in pulse attack, and multiple packet drops leading to RTO are imposed on TCP flows by forcing an internal machine to generate flood to instantly fill the buffer of the access router and maintain the filled status for the desired duration.

The flood traffic suitable for the attack is generated by exploiting two known vulnerabilities, duplicate and optimistic acknowledgment (ACK) spoofing which are inherent in TCP congestion control algorithms. Duplicate ACK spoofing represents the situation where a TCP receiver sends a large number of ACKs to the same data packet so that the sender enters into the fast recovery phase and sends new packets in response to the duplicate ACKs. In optimistic ACK spoofing, the receiver sends ACKs to packets which are sent by the sender but not yet reached the receiver and the result of this is that the sender, as in the case of duplicate ACK spoofing, responds to each optimistic ACK with new data packets. In order to initiate the attack, the attacker in the form of a genuine client establishes a TCP connection with a publicly accessible server (like web or ftp servers) in the targeted network. After receiving a data packet from the server, the attacker by spoofing large number of duplicate ACKs forces the sender to perform a fast retransmit and then enter into fast recovery phase. In fast recovery, the sender ejects as many full-sized new packets as the number of duplicate ACKs it receives. These full-sized data packets act as the flood traffic and will fill the router buffer. The typical value for the number of duplicate ACKs is about 50 packets, which is a widely seen buffer size on edge routers.

Once the buffer is filled, the attacker has to maintain the buffer in the filled status for the desired duration to enforce sufficient packet loss to normal TCP flows. For this, new packets are to be added to the router buffer at the same rate as packets are dequeued from the buffer. This is achieved through optimistic ACK spoofing. The attacker first roughly estimates the number of packets that the server has sent in the fast recovery and generates the first optimistic ACK to the highest packet that was sent in the fast recovery phase. The first optimistic ACK shifts the server from the fast recovery mode to the congestion avoidance mode, where at least one new packet is transmitted in response to additional optimistic ACKs. Optimistic ACKs are generated by incrementing the ACK number of the subsequent ACK packets by Maximum Segment

Size (MSS) and the number of optimistic ACK is selected in such a way that the buffer is maintained for the desired duration. For example, if the access link speed of the targeted network is 1 Mbps and the attacker aims to maintain the queue for 120 ms, it sends 10 optimistic ACK at a rate of one in each 12 ms, which is the time required for the router to transmit a packet of 1500 bytes through 1 Mbps link. The attacker then abruptly stops sending optimistic ACK and this will cause the attack flow to timeout. The normal TCP flows (which are the target of the attack flow) through the access router will also experience a timeout around the same time and both the attack and normal flows will resume re-transmission after one second. This will allow the attacker to repeat the process of duplicate and optimistic ACK spoofing.

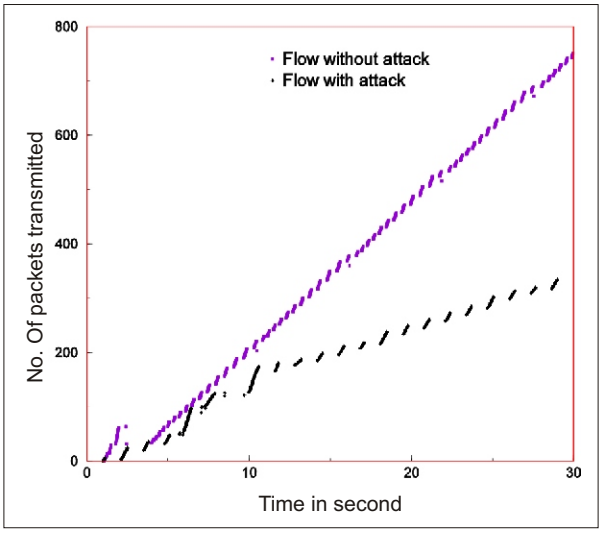

Fig 4.1 Time verses No. of packets transmitted by a TCP flow in presence and absence of the attack flow.

The attack is simulated using network simulator and Fig 4.1 gives the time vs number of packets transmitted by a TCP flow in absence (upper plot) and in presence (lower plot) of the attack flow. The frequent breaks in the lower plot are due to the timeouts imposed by the attacker. It is noteworthy that while the TCP sender in the absence of the attack flow could transmit about 750 packets in 30 sec. interval, this value has come down to 330 in presence of the attack flow. Fig 4.2a, 4.2b and 4.2c show the transmission pattern of the attacker, that of the internal machine and the instantaneous buffer occupancy of the access router respectively during the attack is in progress.

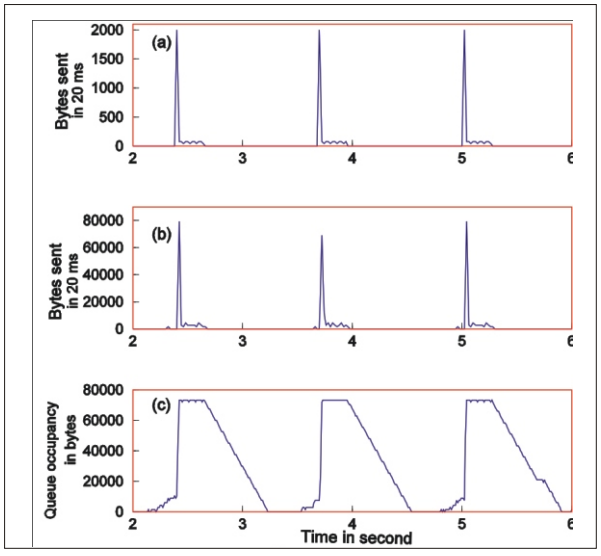

Fig 4.2 (a) Transmission pattern of the attacker, (b) transmission pattern of the internal machine and (c) instantaneous queue occupancy of the access router.

Our work is progressing towards the design and implementation of an Intrusion Detection System (IDS) to detect the attack by passively monitoring the inbound and outbound traffic of the targeted network. The detection is achieved by differentiating maliciously crafted duplicate and optimistic ACKs, which actually forces the internal server to trigger the attack flood. An exclusive signature of the attack, which is being used in our detection system is that the queuing delay introduced to the individual packet which constitutes the attack flood is not always reflected in the RTT of the packet. For each outgoing TCP packet, we define a minimum threshold RTT, $RTT_{min\text{-}thresh}$, before which the packet cannot be acknowledged under normal circumstances. $RTT_{min\text{-}thresh}$ is the sum of queuing delay of the packet at the access router, transmission delay of the packet at the access link and the two-way propagation delay of the access link. The attack is detected whenever the observed RTT of any packet falls below its threshold value.

V Anil Kumar and Dorgham Sisalem

4.2 Active Hop-count Mapping (AHM) for Detection of IP Spoofed DoS Attacks

In this work we propose a method for early detection of Denial-of-Service (DoS) attacks targeted to servers in private networks, which are interconnected using public Internet. A typical

example of such a scenario is a server deployed at corporate headquarters exclusively for serving a set of known clients from branch offices when the headquarters and branches are connected through the Internet. While it is obvious that such servers will be protected with firewalls to block unwanted traffic from sources other than its legitimate clients, an adversary, by IP address spoofing, can pretend to be a legitimate client and launch DoS attacks against the server.

The proposed method is based on the observation that the attacker always spoofs the source address to impersonate the legitimate clients. Hence a server, which can dynamically distinguish spoofed packets from legitimate packets, can take the arrival of spoofed packets as a symptom of DoS attacks. Hop-count (number of intermediate routers between the server and client) information of each client derived from the TTL (Time-to-Live) field of IP datagrams of the clients is used to distinguish a spoofed packet from the normal packet. The server maintains an up-to-date table consisting of the IP addresses of its clients and their hop-count. The mapping is performed actively by sending ICMP requests to all the clients and extracting the TTL of the ICMP replies from the clients. The server compares the hop-count value of incoming packets with that in the table and if these values do not match, the packet is identified as a spoofed one.

The inherent complexity in Internet architecture poses various challenges in making this simple concept to work in reality. First, the initial TTL value of a host depends on its Operating System (OS) and higher level protocols like TCP, UDP etc. and no prior assumption on its value can be made. Second, packets from a given source-destination pair need not always follow the same path. Though, Internet paths are strongly dominated by a single route, link failure, routing loops and parallelism in Internet can cause different packets from same source-destination pair to have different hop-count values. Finally, an attacker can also attempt to spoof the TTL in addition to the source IP address. While the above points make it obvious that the Active Mapping is not practical for generic Internet servers, it can be applied to servers with known client sets. First, the ambiguity in determining the initial TTL under high heterogeneity (in terms of OS and protocol) can be fixed by adopting mutually agreed and common configuration policy. All entities expected to participate in communication can be configured with fixed initial TTL value. Further, adopting an initial TTL value of 255 can significantly limit the TTL spoofing ability of the attacker. Since the hop-count information of clients is obtained and updated at regular intervals of time, any change in topology leading to a different hop-count value will be visible to the server.

In order to evaluate the effectiveness of the detection system, we classify the attackers into three categories. 1) External attackers who may or may not have the topology information: An attacker is considered external if its hop-count from the server is more than the hop-count range of all the clients. 2) Internal attackers without topology information: These are attackers whose hop-count falls in the range of the hop-count of clients, but do not have topology information to find out the hop-count of any clients. 3) Sophisticated attackers: These are internal attackers having topology information so that they can spoof the initial TTL in addition to the source address. The level of sophistication of the attacker can be linked to the fraction of clients for which it can determine the hop-count.

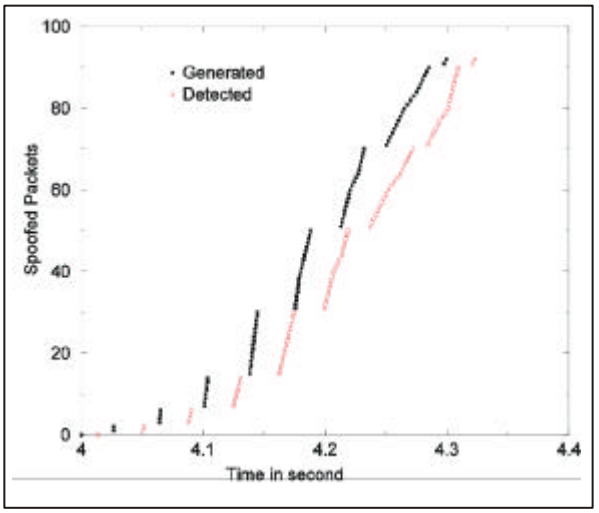

Fig 4.3 Spoofed packets generated and detected verses Time for External attacker

AHM is simulated in network simulator using a topology consisting of 100 network nodes. One node is designated as the server and 98 nodes as its clients. The remaining node represents the attacker and its position in the topology is changed to imitate different types of attackers. Fig 4.3 shows the sequence verses time plot of attack

packets generated and detected in the case of an internal attacker and detected by the server.

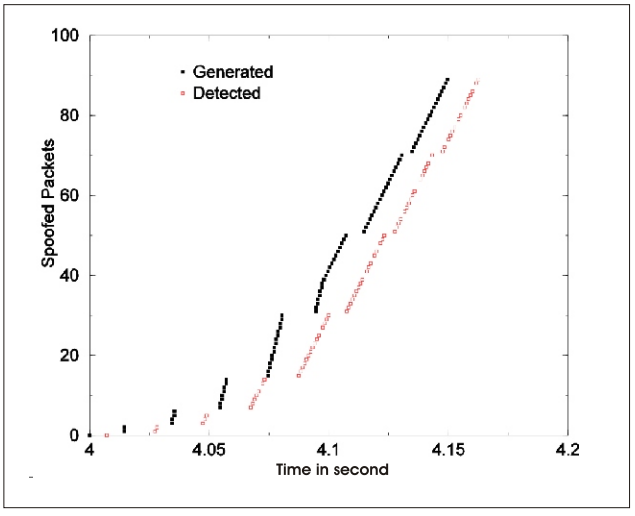

Fig 4.4 Spoofed Packets generated and detected verses Time for Internal Attacker

Fig 4.4 shows the sequence verses time plot of attack packets generated and detected in the case of an internal attackers. As shown in figure, packets 7, 13, 27, 41, 45, 48, 56, 63, 64, 70, 74, 79, 82, 88 are not detected and cause false negative. All undetected packets have spoofed addresses of clients which are equidistant from the server, and this distance is same as the hop-count of the attacker from the server. This shows that if the clients whose addresses are spoofed and the attacker are at the same distance from the server, then the packets generated using these client addresses will cause false negative. However, in real scenarios this false negative is very unlikely to hide the presence of the attacker, because a DoS attack will have large number of spoofed packets and the packets generated by spoofing other clients will reveal the attack. shows the sequence verses time plot of attack packets generated by an external attacker and detected by the server.

In the case of a sophisticated attacker, who successfully computes and spoofs TTL of clients, AHM cannot detect the attack, but can significantly limit the ability of the attacker to distribute the attack source among all the clients. The practicability of such attacks depends on the physical location of the attacker and the fraction of client address for which the attacker can compute the hop-count value. Such attackers can spoof only a fraction of clients whose hop-count is more than or equal to that of the attacker and the attack is likely to exhibit flood nature, which can be identified using other flood based detection techniques.

V Anil Kumar, R P Thangavelu and G K Patra

4.3 Hierarchical Authenticated System using combination of Public and Private Key cryptography

Authentication of participating parties in a cryptographic system is very crucial. Public key cryptography normally takes care of the authentication, as part of the encryption and decryption process, while authentication in a symmetric key cryptography has to be addressed separately. We are proposing a combination of public and private key method to be used as an authenticating system in symmetric key cryptosystems. This is a hierarchical authenticated system, which can locate and uniquely authenticate a person anywhere in the globe. Fig 4.3 shows the basic structure of the authentication system. There is a hierarchy of authenticating agencies each at organization, city, country and international level so that every individual can be located and authenticated. The communication between two levels in the same hierarchy is done using either symmetric or privately-public system, while communication between two levels in different hierarchy is done using Diffie-Hellman algorithm.

In this authentication system the organization authenticating agency (OOA) physically collects the secrete key from the local authenticating agency (LAA). LAA exchange the key with state authenticating agency (SAA) using a dedicated highly secure communication channel. In the similar way each authenticating agency communicates with their respective parent agency using a secure channel. If X from organization OOA1 wants to authenticate Y from organization OOA2 then he will send a request packet to OOA1 with a destination address of **Y@naa2:saa2:laa2:oaa2**. This will reach the international authenticating agency (IAA) through the intermediate agencies. From the destination address IAA decides about, which national authenticating agency (NAA) to be contacted. It will send a pair of random numbers to both NAA1

and NAA2 through its secure channel, which is used by them to generate a key using Deffie-Hellman algorithm. Using the generated key NAA1 communicates a new pair of random numbers to NAA2, which in turn sent to SAA1 and SAA2 through the secure channels of NAA1 and NAA2 respectively. This process is continued till the users generate the secret key using, the pair of random numbers obtained from their OAA's, in Deffie-Hellman algorithm. Any communication using the key can not only provide security but also authenticate the other party.

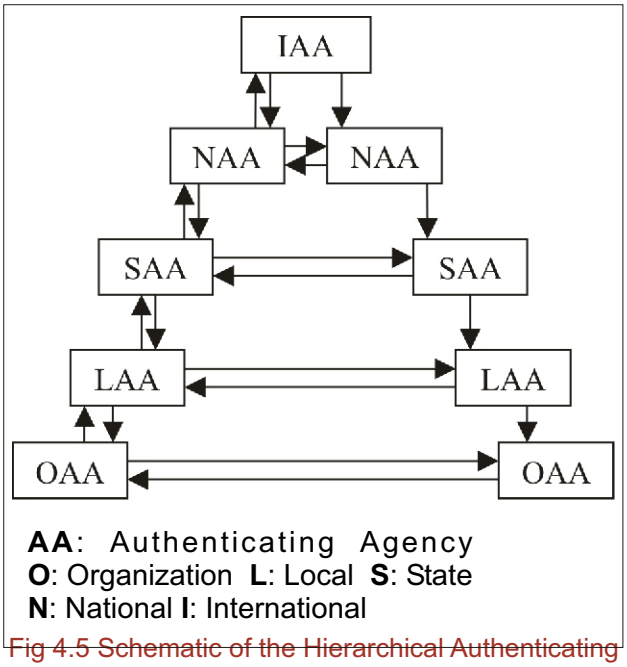

Fig 4.5 Schematic of the Hierarchical Authenticating protocol

The advantage of this authenticating system is that anybody on the globe can be uniquely authenticated and implementation is easy as there are already many organizational systems which follow this tree structure for their administration.

G K Patra and Manas Mukul

4.4 Improved Random Key Generation Algorithm using Conditional Last Bit Flipping

Random Key Generation Algorithm is an interactive public key generation system. In this method two authenticated parties, who wish to exchange information, generate common secret key from two independently generated huge random binary numbers, using multiple, error detection (parity comparison) and elimination cycles. But the major disadvantage of this method is the inability to get key of larger size (frequent 0 bit final keys). This has been overcome by using the conditional last bit flipping (CLBF). So the algorithm is modified as follows (Alice is the sender and Bob is the receiver).

A. Alice and Bob generate Random Binary number string (~10,000 bits) at their respective ends.

B. Both Alice and Bob partition the random number string into blocks of equal length.

C. The parity of the corresponding blocks on both sides is compared.

D. If the parity matches, then the last bit of the block is flipped if the last but one bit is 1. This is called Conditional last bit flipping (CLBF).

E. If the parity does not match, then that block is subjected to binary search. The block is partitioned into two equal halves. Then each of the corresponding partitions on both sides is compared for parity. The sub-block for which the parity does not match is further subjected to the binary search, while the other sub-block undergo step D. This binary search is done till they end up with a two bit pair, which is then discarded.

F. If all the blocks are compared, the bit positions are jumbled, with mutually agreed swapping. Then step B is repeated till the entire block parities match for a number of consecutive iterations.

G. Alice and Bob, verify the key by randomly choosing some bit positions and comparing the bits. If the bits match 100%, then the compared bits are discarded and the rest of the bits are used as the key. If, even a single bit does not match, the whole process is repeated again.

This modification to the algorithm not only increases the probability of getting a larger size key but also generates key successfully, even if the initial percentage matching is very low. The Fig 4.6 shows size of the secret key generated for different size of initial percentage bit-wise matching and for different initial binary number

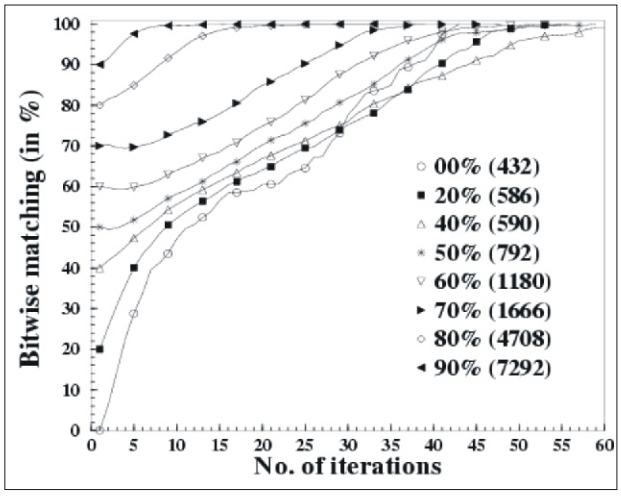

Fig *4.6* Size of the secret key generated for different size of initial percentage bitwise matching and for different initial binary number size.

size and Fig 4.7 shows improvement in percentage matching for different initial percentage matching after every iteration.

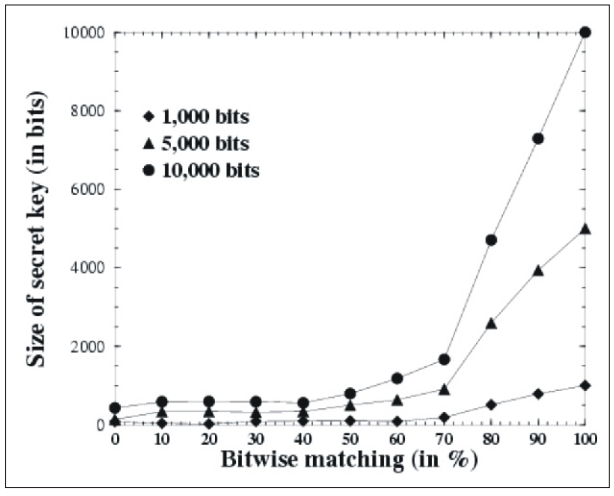

Fig 4.7 Improvement in percentage matching for different initial percentage matching after every iteration. The value in the bracket shows the size of the final key obtained

G K Patra, V Anil Kumar and R P Thangavelu

4.5 Development of a parallel code (version) of AGCM.

Based on the requirements of the NMITLI project on Monsoon Related Meso-scale Forecasting, a parallel version of LMDGCM3.3 has been developed using the domain decomposition method. This version is MPI2 compatible and was tested on the SGI IRIX platform for 8 processors. However, it has been noticed that there is no significant improvement on the performance of the parallel code and efforts are still on to improve the code in terms of scalability.

G K Patra, V Anil Kumar and R P Thangavelu

4.6 Porting of the GCM on to NMITLI platform

The LMDGCM model was ported to the NMITLI computing platform developed by NAL as a part of the NMITLI project on Monsoon Related Meso-scale Forecasting. The sequential version of the code has been successfully ported on to the platform. The new platform runs on Linux operating system and uses Absoft compilers. Modifications to the code were carried out to remove the UNIX dependencies and make it compatible with Linux based systems. Efforts are on to port the parallel version of the GCM code on to the NMITLI platform.

G K Patra, A Mandal and P Goswami

4.7 Performance Evaluation of Itanium2 based Server

In order to enhance the computational facilities at C-MMACS and also to meet the ever increasing computational requirements for the long range monsoon forecast activities, a benchmarking exercise was carried out to ascertain the suitability of Itanium2 based servers. Intel and SGI helped us by providing with a 2 CPU Altix350 server based on 1.4 GHz Intel Itanium2 processor with 1.5 MB L2 cache, 10 GB memory running 64-bit Linux Advanced Server with Intel compilers for this purpose. The LMDGCM3.3 and MOM4 models were successfully ported on to the system. During the performance evaluation it was observed that the 1.4GHz Itanium2 processor was performing at 1.8 times faster than the existing SGI Origin3000 server, which has 500 MHz MIPS R14000 processor. A similar performance was observed in the parallel version of the MOM model also.

R P Thangavelu, G K Patra and P S Swathi

4.8 C-MMACS High Performance Computing Resources

The C-MMACS high performance computing facility continues to provide 100% uptime service for mathematical modelling and computer simulation in the fields of ocean, atmosphere, earth science and engineering to C-MMACS, other CSIR labs and various other research and academic institutions.

The computing capabilities of the HPC facility have been enhanced by upgrading the SGI Origin 3400 server by adding a 4 CPU brick with 8 GB memory. The server now has 24 R 14000 500MHz MIPS processor with 28 GB main memory and 500 GB fibre channel RAID storage space. The machine is capable of running parallel applications using Shared Memory (SHMEM) or Message Passing Interface (MPI). Mainly the system is used for running sequential models like LMD Atmospheric Global Circulation Model (AGCM) and parallel models like Modular Ocean Model (MOM), Parallel Ocean Model (POP) along with various application software like CFD-ACE, NISA , Matlab etc.

A 1.5 TB IBM high availability file server supports the computing facility by providing online storage to the major servers through Network File Service (NFS). This also has a 7.2 TB tape library for automatic backup facility using Tivoli Storage Management (TSM) software.

The C-MMACS network is upgraded to a Gigabit network by the introduction of a Gigabit switch and the major servers connected to it. The clients connect to the server through 100 Mbps switches distributed over the network. The networking is also extended to the annexe building, and the conference room. Air conditioning and Fire alarm systems are also extended to the proposed Cybrary (Cyber Library). A Remote Access Server (RAS) server is installed to enable scientists and other research staffs to access the computing facilities from their residences through telephone connection. A Compaq N800C Laptop was purchased to be used by the C-MMACS staff for presentation in conferences and seminars.

C-MMACS continues to provide e-mail facility to more than 800 users from about 6 nodes distributed over the campus wide network, through its 64 Kbps gateway. Internet service is provided to the C-MMACS users through the 64 Kbps connection to ERNET and 2 Mbps link from NAL to VSNL. The C-MMACS website continues to provide more and more information with most of the internal reports now made available in PDF format. The major attraction of the C-MMACS website is the various activities of C-MMACS, specially the monsoon forecast every year. A website with online registration facility was designed and maintained for the International Conference on Scale Interaction and Variability of Monsoon (Sivom) jointly organised by C-MMACS and DAS-CUSAT at Munnar. A similar website was designed for the North-East conference organised by C-MMACS and Tezpur University.

Application software purchased or upgraded during this year.

A. Matlab upgraded to Version 5.3
B. CFD-ACE+ upgraded to version 2003
C. GAMIT and GLOBK upgraded to version 11.0
D. FEMLAB was purchased for FEM Analysis

The table shown below gives the list of available software at C-MMACS under various platform and categories.

Mathematical Libraries

Complib	High- performance math libraries	SGI
DXML	Extended mathematical libraries	DEC
IMSL	Comprehensive library for numerical and statistical analysis	SGI
NAG	Numerical and statistical analysis	SGI
NUMERICAL RECIPES	Software for numerical analysis	SGI, Intel
SCSL	SGI Cray scientific library	SGI

Application Packages

Biology & Chemistry

AMBER	Modelling of peptides / nucleic acids / carbohydrates	SGI
DeFT	Gaussian density functional program	SGI
deMon-KS	Molecular orbital solution of the Kohn-Sham DFT system of equations	SGI
PCMODEL	Molecular modelling	SGI

CAD/CAE

CAMAND	Computer aided modelling, analysis, numerical control, design and documentation	SGI
CFD-GEOM	Surface modelling and grid generation	SGI
SDRC I-DEAS	Solid modelling	SGI

Earth Sciences

BERNESE	GPS data processing	SGI
GAMIT	GPS data processing	SGI
GLOBK	GPS data processing	SGI
MOM	Global ocean circulation (Modular model)	SGI
LMDZ	Atmospheric Global Circulation Model	SGI
TIDAL	Shallow water simulation and pollutant transport	SGI, Intel
SEISAN	Seismic Analysis	Intel

Fluid Flow, Heat and Mass Transfer

CFD-ACE+	Computational fluid dynamics	SGI
NISA	Finite element fluid dynamics	SGI
PHONENICS	Computational fluid dynamics	SGI
PORFLOW	Porous media flow, heat and mass transfer	Intel

Scientific Visualisation

CFD-VIEW	Graphics for CFD	SGI
Ferret	Visualisation tool for atmospheric and oceanic applications	SGI
GrADS	Graphical display for atmospheric and oceanic applications	SGI, DEC
IDL	Interactive data analysis and visualisation	SGI
NCAR Graphics	Advanced graphics display and mapping	SGI
SigmaPlot	Data manipulation, regression and curve fitting	Intel
SigmaScan Pro	Image digitising software	Intel
TableCurve 2D	Automated curve fitting and equation discovery	Intel
TableCurve 3D	Automated surface fitting and equation discovery	Intel
TECPLOT	General purpose 3-D graphics	SGI, Intel

Structural Mechanics

NISA	Finite element analysis	SGI
SDRC I-DEAS	Finite element modelling	SGI
FINEART	Finite element modelling	Linux
FEMLAB	Finite element modelling	SGI

Miscellaneous

ACRPLOT	General purpose plotting package	Intel
AXUM	Technical Graphics and Data Analysis	Intel
FLOWPATH	2-D flow and contaminant transport in sub-surface	Intel
GMT	Generic Mapping Tools	SGI
GNUPLOT	General purpose plotting package	SGI
IDL	Interactive Data Analysis and Visualisation	SGI
MACSYMA	Applied Mathematics software	Intel
MATLAB	Mathematical and symbolic computation	SGI
MathCAD	Mathematical calculation, visualisation and documentation	Intel
MODFLOW	3-D simulation of flow in sub-surface	Intel
PdEase	Applied Mathematics software	Intel
SCILAB	Mathematical and symbolic computation	SGI, Intel
SPSS	Advanced statistical analysis	DEC
TSM	Tivoli storage management software	IBM
Visual MODFLOW	3-D flow and contaminant transport in sub-surface	Intel
Visual Studio	Microsoft Development Tool.	Intel

The following procurements have been initiated during the last year and installation is in progress

A. Purchase order has been placed for a 12 processor, Itanium2 based SGI Altix 350 server, with 60 GB main memory and 1TB RAID disk storage.
B. Purchase order has been placed for 20 numbers of Pentium 4 based Linux desktop systems.
C. Purchase order has been placed for 1.5 TB additional RAID storage to the SGI Origin 3400 server.
D. Purchase order has been placed for Pbs-Pro (Work load management), Total View (debugger) software.

Technical and computing supports were provided for various courses, workshops and conferences organized by C-MMACS. Training was provided on basic Unix and GAMIT/GLOBK installation on Linux to the participants of the SAARC GPS training course. Technical assistance was provided to various divisions of NAL for troubleshooting the Linux based E-mail servers. Ph. D. students from IIT Delhi and Gandhigram Rural Institute were provided with high-end computing and visualization facilities. Training and computing facility were provided to the students of various engineering colleges and universities, to enable them to carry out their academic project works. Technical and design support was provided for designing the AMS-India conference, web page, with online registration facility, organized by ISI Bangalore and IISc. Bangalore.

R P Thangavelu, G K Patra, V Anil Kumar, N Prabhu and Seenappa