Resume

Name	Ankur S Jain
Address	704 Campus Dr #8C, Stanford CA 94305
Telephone	650 497 6801
Email	ankur@cs.stanford.edu

Academic Status	Masters:	Stanford University (Sep 96 -), Major: Computer Science GPA: 4.05
	Bachelors:	IIT Delhi (July 91 - June 95) Major: Computer Science & Engineering GPA: 9.43 / 10 (3.72 / 4.0)

Areas of Interest **1.**Networking & Distributed Systems
2.Operating Systems

Employment Details

Company	AT&T Bell Labs (From Sep 1, 95 to Sep 20, 96)
Designation	Technical Staff Member - Computer Engineer
Department	Networking Research Department
Center	Networking and Distributed Systems Research
Nature of work	Development, enhancing and porting of a native ATM stack Achieving end-system quality of service in networks

Company	Visigenic Software (July 7, 97 to Sep 23, 97)
Designation	Software Engineer - Summer Intern
Nature of work	Development of Security module for CORBA.

Publications

- Ankur Jain and S. Keshav, Achieving Service Quality in End-Systems (submitted for publication)
- Ankur Jain and S. Keshav, Native-mode ATM in FreeBSD: Experience and Performance, *Proc. NOSSDAV'96*, April 1996.
- Ankur Jain and Navin Mittal, Signaling Issues in ATM Networks, *Undergraduate Thesis, Dept of Comp Sc and Engg, IIT Delhi*

Academic Honors

- Won the first prize in the novice category in the Student VLSI Design Contest 1997 at the University of Utah (for the RSA chip designed in EE272a)
- Awarded the merit prize thrice in IIT Delhi for obtaining the highest semester GPA
- Ranked 3rd in State Level Exam for selection of candidates for the final National Talent Search Exam from a pool of 30,000 students.
- Awarded the National Talent Search Scholarship in 1989.

Programming Languages	Proficient in C++, C, Verilog Familiar with Java, Perl
Other Skills	Kernel Programming in FreeBSD and Linux

<u>Coursework</u>

Masters (upto Spr 96)

Intro to Comp Networks(TA'd)	Introduction to VLSI Systems	AI: Principles & Techniques
Advanced Topics in OS	Design Projects in VLSI	Deductive Database Theory(CS345)
Distributed Systems	VLSI Systems:Testing & Simulation	Intro to Computer Graphics
Multimedia Networking & Communications		

Bachelors (CS courses)

Compilers	Digital Integrated Circuits	Signals & Systems
Systems Programming	Computer Architecture	Data Structures
Data Communications	Digital Hardware Design	Discrete Structures
Introduction to OS	Microprocessors	Theory of Computation
Introduction to Databases	Analog Interface Design	Analysis & Design of Algorithms
Concurrency Control in Databases	Advanced Computer Architecture	Digital Image Analysis
Computer Networks	CAD of Digital Systems	Numerical Analysis

Work at Bell Labs

I worked at AT &T Labs for a year, and was involved in porting, testing and enhancing of Idlinet, a native mode ATM stack.

- **Porting of ATM-stack on FreeBSD**

  The aim of the project was to port the ATM stack (implemented on PCs running the Brazil operating system) on PCs running FreeBSD and connected to a network of fore-ATM switches. The ATM stack that was ported has three main parts namely the application entity, signalling entity(which takes care of the establishing/closing connections, allocating resources) and the transport entity. The first two entities run in the user space while the transport entity runs as part of the kernel. The project involved rebuilding of modules for communication between the three entities, and changing the part of the stack running in the kernel space.

  The aspects that had to be dealt with while porting were different interprocess communication mechanisms in Brazil and FreeBSD, static allocation of pseudo devices and their management, abscence of locks in FreeBSD which required building of a lock module featuring reader-writer locks and semaphore locks, building of a sort of error handling stack used by the ATM-stack but unavailable in FreeBSD which involved in dealing of processes coming out of locks forcefully due to signal delivery, the difference in the clock granularity of the timer interrupt, page-allocation of kernel memory in FreeBSD as opposed to physically contiguous allocation in Brazil, etc.

- **Device Drivers for ATM cards**

  The ATM-stack supported FORE ESA-200 EISA card, but it had to be moved to systems having FORE PCA-200E PCI and Zeitnet 1225 PCI cards. So the next phase of the work involved writing device drivers for these two cards initially for FreeBSD.

- **Porting the stack to Linux**

  The large community of linux users prompted the move for porting the ATM-stack to linux. The porting from FreeBSD to linux was not as difficult as the one from Brazil to FreeBSD. Other than the ones described in FreeBSD porting, the new aspects that had to be dealt with were the different support for the device driver probing, initialization and registration, the different memory management and allocation schemes, abscence of sleep for a certain time period, etc.

- **More OS support for QoS**

  The guaranteed service in ATM makes some demands on the network and OS resources. The ATM-stack Idlinet supported guaranteed service, but the support used to crumble in the onslaught of numerous besteffort applications with the besteffort applications taking away a big chunk of the resources from guaranteed service applications.

  To get around this problem, t he task-scheduler which is a part of the ATM-stack was modified to control the resource use when scheduling the tasks for the various services, the memory management module was changed to take care that guaranteed service applications had enough buffer space and that the best effort applications did not take up all the buffer space. The leaky bucket support on the card was used to block the best effort applications from sending too much data on the link.

B.Tech Major Project - Signalling Issues in ATM networks

IDLInet is a native mode ATM protocol stack built in IIT, Delhi. The project was to enhance the protocol stack. The project dealt with Signalling Issues and upgrading the stack by putting additional features like recovery from link and node failures, and expeditious rerouting of the connections, which is essential to make ATM networks suitable for telephony.

ATM offers a connection-oriented point to point service. For this a Virtual Connection is established between the communicating nodes and all communication takes place using the Virtual Connection. So, when a link goes down one needs to tear down the Virtual Connections and re-establish communication between the end nodes using an alternate route and when a link comes up the main task is to re-establish the *signalling channel* (the Virtual Connection reserved for signalling between the nodes at the ends of the link). The project involved the following tasks

- **Tearing down the Virtual Connections**
  On link failure, the nodes at either end of the link identify the Virtual Connections using that link and initiate a crash mode teardown process by sending crash messages along the Virtual Connections. The teardown process is to be carried out on Virtual Connections in all possible states (Connection established, Connection being established, renegotiating resources, Connection being closed, Connection undergoing crash teardown) which involves chasing the messages travelling on the signalling channels and stopping the messages coming from the opposite direction on the signalling channels.

- **Link recovery**
  When a link comes up, the nodes at the two ends re-establish the {\it signalling channel} and the {\it hello channel} (the Virtual Connection used to exchange topology information)

- **Rerouting of Virtual Connections**
  The end nodes of the connections, on receiving the crashmessages, wait for the hello packets (carriers of topologyinformation) and on their arrival initiate a establishment sequencealong another route if the network is not partitioned. The reroutingprocess preserves the connection id\'s at either end and so it istransparent to the application layer except in case of guaranteedservice where the application is informed about the crash and theinability of the system to satisfy the service guarantees duringthe period when the rerouting process is going on. Timeouts are usedto safeguard against the failure of arrival of hello packets and rerout messages.

The recovery routines can handle <u>multiple link and/or node failures</u> along a connection and also the <u>failures during the recovery process</u>.

Guide : Dr. Huzur Saran, Asst. Professor, IIT Delhi

VLSI Design - Self timed RSA encoder/decoder

The project was done as a part of the class EE272A (Design Projects in VLSI) at Stanford University. The project aimed at the design of a modest sized CMOS chip, which included writing a functional model (in Verilog), using synthesis tools, custom layout, and simulation.

RSA coding involves two parts, key generation and the actual decoding/encoding which is essentially modulo exponentiation where the exponentiator and modulo are large primes. The project aimed at building a self-timed scaled down version (8-bit version, actual keys are of the order of 512 bits) of this module. The exponentiation involved multiplying iteratively and the multiplication involved an iterative shift-add loop (tree multiplier could not be used because of the modulo with prime number).

The inner multiplier loop was implemented as a self-timed token ring, with the clock for any stage being derived using the ouputs of the input consumed and completion detectors at the previous and next stages. For completion detection, dynamic dual rail domino-logic was used, which posed multiple challenges, the most important being the doubling of the number of wires (to carry the dual rail signals) while laying out the datapath and charge sharing. An attempt was made to get a zero-overhead design (where the token does not have to wait for the clock signal at any stage) , however it could not be implemented because of lack of space on the chip (2.2mm x 2.2mm, 2 micron technology).

More details of this project can be obtained from http://www-leland.stanford.edu/~ajain/chiprep.ps

Security for CORBA

This work was done during as part of internship at Visigenic Software during the summer of 97.

The task involved implementation of parts of the CORBA 2.0 security specifications concerning authetication (getting the credentials from the principal authenticator), granting of rights for specific attributes on credentials, setting the required rights on a particular operation on an object and access control for the operations .

Other projects

Parallel Poisson Solver

The project involved development of a pipelined version of the Poisson Solver executable on the Cosmic Environment and Parallel Virtual Machine. The code involved iteratively computing the values of elements in a matrix using column by column relaxation. This line relaxation technique enabled efficient pipelining of the code. Blocking receive calls were used to synchronize the various node processes in order to print intermediate and final results. In course of the project, I also developed a general purpose high level pipelining procedure to pipeline iterative algorithms using line relaxation for matrix computations. Using this, the user simply needs to supply the initial guess to the solution, the boundary conditions, desired tolerance of the solution and the function to compute the new values for the elements of a column using the old values of itself and its two neighbouring columns. The rest is handled by the procedure and the user gets the desired solution.

Guide : Dr. Gautam Shroff, Asst. Prof IIT Delhi
Project work was done during the sophomore year

Parallel Continuity Equation Solver

The aim was to create a parallel version of a program for computing flow field in a cavity with or without thermal and/or solutal fields, with the same structure as that of the sequential code.The code involved solving continuity flow equations equations iteratively in spatial domain with corrections occuring at each time increment.

Guide : Dr. Gautam Shroff, Asst. Prof IIT Delhi
Project work was done during the junior year

FFT for INTEL i860XR

(Done as part of the Industrial Training at Center for Development of Advanced Computing (CDAC), Pune from May 12, 1994 to July 6,1994.)

The project involved development of code for general radix real FFT in the assembly language of the processor*INTEL i860 XR* (the cpu for *PARAM 8600*). The i860 XR allows upto 3 instructions to be executed in a single clock cycle,one integer core instruction and two floating point instructions (one floating add and one floating multiply instruction in the two different floating point pipelines). The aim was to code the FFT routine(a computation intensive part of the Weather Project code) in such a way so as to fully utilise the on-chip cache and the the parallelism between the integer core,floating point add and floating point multiply units of i860XR.

Guide : Dr. Sudheendra Hangal, MTS at CDAC, Pune,INDIA.