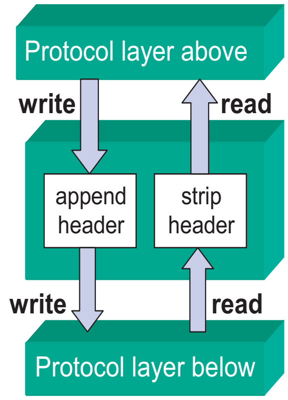
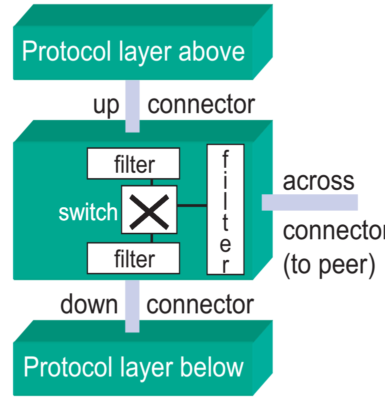
A grand challenge in computer networking

Network management —what could be less interesting? The phrase evokes images of hapless IT workers poring over manual pages and typing rows of cryptic management commands, while pulling out their hair trying to figure out why Fred can't connect to the server but Kathy, in the next office, can.



Historical protocol layering abstraction for data delivery



Is there an analogous layering abstraction for network management?

Network management has always been viewed as a black art, understood only by gurus. This guru-centric view hurts most in two areas: first, settings where even the best gurus aren't good enough to control all situations that might arise (such as critical infrastructures requiring reliability measured as "five-nines" or more), and second, settings where there are no gurus at all (home networks). Networks whose management is based on gurus cannot evolve much beyond the abilities of the available gurus, and networks that grow too complex become less reliable, less secure, and more expensive to operate.

Network management is all about discovery: discovery of boxes and discovery of protocol layers supported by those boxes.

Self-managing networks are the obvious solution, but how to build such a network is far from obvious. IBM, which calls this grand challenge *autonomic computing*, has not reported much progress despite a now three-year corporate focus. The UPnP (Universal Plug and Play) industry forum defines standards for self-configuring devices and PCs, but the standards for basic connectivity alone cover 500 pages!

Why is the construction of self-managing networks so difficult? CS professor Paul Francis believes it is because there is no real architectural framework for network management. Architects of the early Internet devised routing protocols only to hold the network layer together. As a result, every vendor was left to implement a distinct style of management. "Network management research is like doing a 5000-piece jigsaw," says Francis. "There are many interesting sub-problems, but putting the whole puzzle together is more about piecing together hundreds of evolving details and less about elegant solutions." Not surprisingly, many networking researchers have dismissed the problem area as hard but dull.

CS Professor Paul Francis: Who would've thought that network management might be interesting after all?

The Internet has always had an extensible architectural framework for data delivery services: the protocol layering model, which today is taken for granted. The simple idea that each layer adds and strips its own header, and provides basic services to the layer above, is arguably responsible for much of the tremendous expansion and evolution of the Internet over the last two decades. Is there an analogous extensible framework for network management? Francis believes there might be and that it could lead to a new generation of self-managing networks.

Network management is all about discovery: discovery of boxes and discovery of protocol layers supported by those boxes. Historically, automatic network management protocols performed discovery using the data delivery services provided by protocol layers. "What if," asks Francis, "in addition to supporting data delivery, each protocol layer had some native support for discovery? What would this support look like, and how would it be used?"

All protocol layers appear to have a small number of fundamental structural components in common. These components include connectors to layers above and below as well as across to peers in other devices. They also include filters at each end of the connectors and internal switches that join the connectors together. Given this structural commonality, there should be a small number of operations to manipulate those common components: discovery of potential or realized components, connection of components, and testing of components. With the right set of abstractions, it should be possible to hide most of the gory details associated with the management of a protocol from network managers (human or machine). The network now can be seen as a simple (albeit large) graph of nodes, links, and filters, rather than as thousands of obscurely interrelated protocol parameters.

"This network management solution isn't going to be a quick hack like NAT was," says Francis, "although I hope its impact is at least as dramatic." Network Address Translation (NAT), invented by Francis in the early 90s, is widely acknowledged as having averted the IP address shortage crisis, allowing the Internet to expand beyond its role as an experimental research network. NAT allows a router to act as an agent between the Internet and a local, or private, network, so that a single unique IP address can represent an entire group of computers. If you have a local network in your home, thank Francis for developing NAT, which made your network possible.

"Who would've thought that network management might be interesting after all?" quips Francis. That Francis did is what makes him the networking leader he is today.

Computing in the arts

The influence and potential of computer science extends to the realms of the imaginative and the aesthetic. To encourage students to learn about this role, CS and CIS have worked with faculty in other disciplines to create an undergrad minor or concentration in the College of Arts & Sciences, called Computing in the Arts. This minor is the latest example of how we are expanding opportunities for students by working with forward-thinking professors across the campus.

The concentration in Computing in the Arts:
Requires CS 165 and five courses in one track (which may include two from another track to encourage interdisciplinary study). The topics of courses, given below, illustrate some of the infiltration of computing into the arts at Cornell.

Computer Science Track
Visual Imaging in the Electronic Age
Computer Game Design
Computer Graphics
Artificial Intelligence
Natural Language Processing
Machine Learning
Computer Animation
Data Mining
Human-Computer Interaction Design
Language and Technology

Music Track
Computer Game Design
Digital Music
Computers in Music Performance
Scoring the Moving Image
Sound Design and Digital Audio
Digital Performance
Improvisational Theory
Counterpoint
Composition in Recent Styles
20th-Century Musical Languages
Physics of Musical Sound

Psychology Track
Visual Imaging in the Electronic Age
Computer Graphics
Cognitive Psychology
Digital Music
Visual Perception
Auditory Perception
Psychology of Music
Human Perception: Applications to Computer Art and Visual Display

CS faculty member Graeme Bailey, educated as a mathematician and also as a performing musician, piloted the project, along with Steve Stucky (Music, composer, winner of a 2005 Pulitzer Prize) and Carol Krumhansl (Psychology, audio and music perception). The minor currently has tracks in music, psychology, and computer science. A parallel minor in Digital Arts has been approved in the College of Art, Architecture, and Planning.

Supporting the minor is Bailey's CS 165 course on Computing in the Arts. An innovative mix of formal lecture and studio work, with minimal prerequisites, the course focuses on ideas rather than software packages. Working with poetry, visual art, sculpture, and music, students learn about randomness and stochastic processes, symmetry, structure and group theory, dynamical systems and embedded structures, shape, deformation, and topology —applying all this to actual artistic creations. Future versions will have enhanced perception/cognition content. In the studio environment, students learn to critique each others' work in constructive ways. Creativity is an integral part of the course.

Cornell is putting resources into research in computing in the arts. Music has hired faculty in computer music. Psychology has freed resources so that Krumhansl can co-teach CS 165, imbuing it with her insights of a lifetime's work in audio perception. Fine Arts has committed to hiring in digital art. Dance is planning productions involving computer response to visual capture.

Various faculty are distilling their active research into presentations aimed at both undergrads and experts from other disciplines. For example, CIS visiting faculty and CS PhD Fabio Pellacini, previously of Pixar Animation Studios, has given talks on making high-level computer graphics artist-friendly (used centrally in making *The Incredibles* and a new film, *Cars*). A week in April 2004 was *Digital Arts Graphics Week*, with the President and Vice President of Pixar, Senior Vice President of Sony Imageworks, and Mark Levoy of Stanford giving presentations (many of the participants were Cornell graduates). For two weeks in Sept. 2004, Cornell hosted *Perspectives on Digital Music in the 21st Century*, with major composers and leading academics from the US and Europe exploring the past and future of many contributions of computer science in the realm of music and sound.

CS moves into 22,000 additional sq. ft. of new space constructed on top of Upson Hall.

Former PhD student Kurt Mehlhorn and frequent visitor Wolfgang Paul receive the German Leibniz Prize.

David Gries publishes the first of five years of Taulbee Surveys, which give data on PhD-granting departments. The five years of surveys have an almost 100% completion rate.

Bruce Donald, Dave McAllester join.
John Hopcroft becomes Chair.
CS grows to 25 faculty members and 200 computers.

1987

David Gries chairs the Computer Science Board, the precursor to the Computing Research Association (CRA). This Board was formed in 1972 to provide a forum for the discussion of issues in research and education in computer science.

John Hopcroft chairs the NSF Advisory Committee for Computer Research.

Gerry Salton receives the Distinguished Science Award from the Humboldt Foundation. This foundation, created by the German government in 1953, enables scholars to do research in Germany.

Don Greenberg receives the ACM Steven Coons Award. This highest award in graphics honors lifetime contributions to graphics and interactive techniques.

Ramin Zabih and David McAllester receive the Best Paper Award at the AAAI Conference. McAllester is now Professor and Chief Academic Officer, Toyota Technological Institute at Chicago.

Devika Subramanian,
Dan Huttenlocher join.

1988

Juris Hartmanis and John Hopcroft are elected to the National Academy of Engineering.

Gerry Salton is named a Pioneer of Computing in the *Annals of the History of Computing*. He receives the ACM Award for Best Review in Computing Reviews.

Don Greenberg receives the National Computer Graphics Association Academic Award.

Eva Tardos receives the 1988 Fulkerson prize for the paper *A strongly polynomial minimum cost circulation algorithm*.

Fred Schneider takes over as Editor-in-Chief of *Distributed Computing*, and David Gries becomes a Managing Editor *of Information Processing Letters*.