RELIABLE MULTICAST CONGESTION CONTROL (RMCC)

Joseph P. Macker
Information Technology Division
Naval Research Laboratory

R. Brian Adamson
Newlink Global Engineering Corp.

ABSTRACT

Abstract -- At present, there are no standardized, Internet-based multicast transport protocols that provide effective, dynamic congestion control methods for safe, widescale deployment of end-to-end rate adaptive applications (e.g., file transfer). Recent research and standardization efforts are beginning to address these issues. This paper describes ongoing research and development related to network congestion control mechanisms for multicast data transport. We concentrate on the design issues for reliable multicast protocols in particular and we describe our approach for adding dynamic congestion control to a negative-acknowledgement oriented protocol. We also present simulation and modeling results demonstrating prototype system performance, including analysis of TCP fairness and router congestion indicators. We also relate our work to other ongoing research and standards development efforts.

PROBLEM BACKGROUND

The successful day-to-day operation and proliferation of Internet Protocol (IP) technology worldwide has been in a large part due to the existence and wide scale use of a standardized, reliable unicast transport protocol, the Transport Control Protocol (TCP) [RFC793]. In addition to providing for reliable data transport, TCP also provides effective, end-to-end congestion control mechanisms [Jacobson88, Stevens97]. At present, multicast transport solutions lack such standard, proliferated approaches to congestion control that operate robustly across a wide range of network scenarios and support fairness when deployed alongside existing network transport standards.

Best effort (i.e., unreliable) native IP multicast transport service presently supports a number of useful real time multimedia and videoconferencing Internet applications. Unlike these applications, reliable multicast (RM) applications are likely to support more computer-to-computer automated applications with little or no human monitoring of quality or congestion levels. An overview of RM application areas and design issues applicable to military internetworking was covered in previous literature [MCK96]. It is clear that applications in this more rate adaptive design space must be proficient at discovering maximal effective network capacity and avoiding undesirable congestion behavior with competing network flows. Therefore, adequate congestion control mechanisms are a key requirement for widespread standardization and deployment of RM solutions and applications.

Another specific concern for the Internet community is the impact RM traffic has on other Internet traffic (particularly TCP flows) during times of congestion [MRBP98]. In addition, reliable multicast congestion control (RMCC) design requires special attention since the congestion-related impact from a single traffic flow can easily span a wide area of the topology.

RMCC APPROACHES

Within the Internet protocol community, end-to-end protocol design and operation is often considered an important design goal to minimize the reliance on specialized network component intelligence and/or signaling. This often means a more closed loop protocol design that operates without specialized network component support. We primarily focus on such an end-to-end protocol design framework in this paper. However, we do consider emerging router and packet forwarding technologies that may further enhance performance. These mechanisms may also further improve wireless operation of Internet-based end-to-end protocols. Some promising technologies include advanced queueing strategies (e.g., random early detection (RED)) and explicit congestion notification (ECN) techniques [FF99].

It should also be mentioned that open loop proactive traffic management strategies (e.g., weighted fair queueing) can improve the overall performance of end-to-end congestion control protocols and we studied these techniques in the past for queue separation of transport types (unicast vs. multicast) and for class-based bandwidth management [MC97, MP98]. Again, while these components enhance performance, it is desirable to design effective end-to-end congestion control mechanisms that can operate in the absence of these enhancement technologies. Network management is also simplified by having end-to-end transport mechanisms as a core service.

In this paper, we focus primarily on end-to-end congestion control research and software development done within a prototype NACK-oriented reliable multicast protocol framework. Due to the interest in general Internet use and the desire to maintain low deployment and management complexity, another

This work was supported in part by the Office of Naval Research (ONR).

goal of this work is TCP fairness analysis. We discuss these protocol-related design aspects and provide comparative simulation results, including some analysis with the use of RED/ECN extensions. We also present early results of fairness using selective acknowledgement based TCP (TCP-SACK).

STANDARDS DIRECTION

At the time of this writing, the primary Internet standards body, the Internet Engineering Task Force (IETF), is developing standard building blocks and protocols for reliable multicast within the Reliable Multicast Transport (rmt) working group (WG) [RMTWG]. The initial goals of the rmt WG include the development of three standard RM protocol areas.

- NACK-oriented RM (NORM)
- Tree-based Acknowledgment RM (TRACK)
- Asynchronous Layered Coding RM (ALC)

We are contributing to the work in progress of the RMT WG and are helping define approaches for the NORM protocol area. The NORM protocol design will include a number of protocol components: NACK processing and suppression, FEC-based multicast repairing [Macker97a-b], and congestion control processing. A component overview of a present NORM protocol design is shown in Figure 1 [ABFHM00]. This paper will focus more specifically on our work done in the area of congestion control operation.

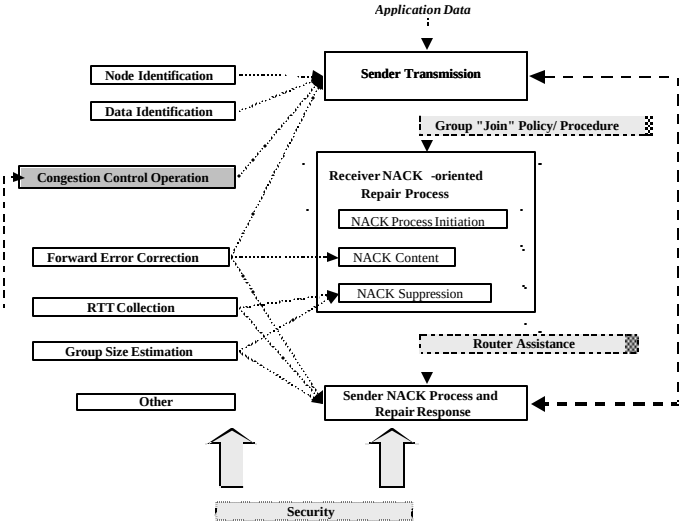

Figure 1: NACK-Oriented RM Design Components

MDP RMCC DESIGN AND EXPERIMENTATION

In our recent RMCC research, the Multicast Dissemination Protocol[1] (MDP) is serving as the software design framework for prototyping RMCC extensions. The MDP design is already in use in a number of DoD networks and applications and provides end-to-end reliable transport of data over IP multicast capable networks. MDP provides efficient, scalable, and robust bulk data transfer (e.g., computer files, transmission of persistent data) capable of operating in a range of heterogeneous networks and topologies. MDP also provides a number of different reliable multicast services and modes of operation. The protocol details and associated software toolkit are documented elsewhere [MA99a]. Present MDP software distributions provide both a fixed rate-control mechanism and an experimental dynamic congestion control option. Fixed rate operation and design is well established and is being applied effectively in operational practice (e.g., VSATs, Internet MBone, ground-based packet radio, etc).

Since MDP is currently used primarily in fixed rate operation, we herein refer to the dynamic congestion control (CC) protocol extension as MDP-CC. MDP-CC is mainly an end-to-end extension of the MDP protocol engine providing congestion sensitive rate-control, with some aspects of protocol behavior modified to accommodate congestion control sensing and feedback. The target rate-control model for MDP-CC is loosely based upon a steady state TCP throughput model [PFTK98]. This previous UMASS research developed an accurate steady state model to estimate the rate at which a TCP source transmits, given roundtrip delay, delay variation, and packet loss as input metrics. This TCP model can be represented by the following functional relationship:

(1) Throughput = f (PacketSize, RTT, TO, P, b)[2]

Congestion control fairness can be defined in a number of ways, but our primary goal here was to initially design a TCP-fair mechanism and we chose a *worst path* fairness model [WC98]. This model defines fairness in terms of a multicast transmission rate compared to the equivalent TCP *worst-case* rate among the different multicast source-receiver routing paths. This conservative approach guarantees similar fairness that TCP uses to compete with other TCP flows within an internetwork, which is an end-to-end path fairness model. MDP-CC operates by using multicast group packet loss and round trip time (RTT) estimation algorithms to estimate worst path model TCP friendliness among the receiver group. We then apply a rate adaptive method of linear increase and multiplicative decrease around a goal rate calculated from the steady state TCP rate model in equation (1).

Figure 1 outlines a number of major functional component relationships within the NORM protocol process. While a number of the RMCC building blocks shown in Figure 1 may be somewhat general and reusable, there are design and integration issues with a NORM protocol requiring specific attention. One of the more important issues is balancing the feedback suppression requirement against the need for timely congestion-based feedback. We have added an additional coordinated group

[1] see http://manimac.itd.nrl.navy.mil/MDP

[2] Throughput = rate in bytes/sec, RTT = round trip time estimate in secs, TO = applicable TCP retransmission timeout, P = fractional loss estimate, b= Number of packets acknowledged by a TCP ACK.

mechanism, discussed later, that addresses these feedback tradeoff issues and may significantly improve scalability of flat protocol operation to large multicast groups.

MDP-CC Parameter Estimation

In order to apply (1) for rate-based RMCC, a number of functional parameter values and estimates are maintained by MDP-CC. The current MDP-CC design uses the source data *segment size* plus the overhead of an MDP data message for the *PacketSize* parameter. Measurements of round trip packet delay *RTT*, delay variation[3], and packet loss *p* are dynamically obtained through both *proactive and implicit* feedback from receivers within the group. For parameter *b*, a fixed value of 1 is assumed in the current implementation. Using the equation (1) with the dynamic parameter estimates, the *worst path* goal rate is established by determining the lowest goal rate among the different source-receiver paths. Using this goal TCP rate and a linear increase/exponential decrease adjustment algorithm, MDP-CC determines available capacity and attempts to fairly share it with TCP or other transport flows (e.g., other MDP flows) with similar congestion-sensitive behavior.

While equation (1) provides an accurate steady-state model across a wide range of conditions, it did not provide any guidance regarding dynamic algorithm design or performance. To address this, we developed dynamic algorithms for protocol operation and estimation and use (1) only as a goal rate for achieving TCP fairness. MDP-CC is designed to maintain a *worst path* fairness model even under dynamic conditions through timely probing of a subset of the receivers to track the current worst *congestion condition* paths within a multicast group. This timely feedback process allows MDP-CC to quickly discover available network capacity and more rapidly adjust its transmission rate in the face of congestion or to take advantage of newly available capacity. The reaction time is designed to be a function of the group greatest round trip time (GRTT). Multicast protocol scaling issues make it prohibitive to excite rapid feedback response from the entire receiver set (which may be quite large), so the source elects a subset of receivers (a default of 5 in the current implementation) with a goal to include the current significant multicast topology *bottlenecks* in the subset. A member of the subset of rapidly probed receivers is called a congestion control *representative* as in [DO98], but the election and reporting process is based upon different criteria than used in this previous work. The *representative* set dynamically changes and is elected based upon *worst path* estimation criteria, a combination of path loss and RTT metrics. We have performed initial simulations of congestion representative migration in multiple bottleneck topologies. From these early results, the representative election algorithm seems to track and manage congestion migration reasonably well. The reader is referred to [MA99b], for further MDP and MDP-CC related design details.

[3] *TO* from equation (1) is a function of *RTT* and delay variation

SIMULATION DESIGN AND RESULTS

Protocol simulation and analysis has been an important part of our RMCC research efforts. Our approach to developing and analyzing simulation models has been to use little or no process abstraction during analyses. For accuracy, we feel that protocol timing and dynamic effects resulting from varying topology and traffic model scenarios require a detailed protocol simulation. At present, the complete MDP software design includes simulation model extensions for both the Berkeley *ns*[4] and OPNET simulation environments. Due the availability of supporting software and additional research tools, *ns* simulation model extensions are presently our preferred choice to investigate RMCC enhancements to MDP and to evaluate TCP fairness.

A graphical example of one of the simulation scenarios we use to examine MDP-CC is shown in Figure 2. Simulation scenarios, such as Figure 2, are easily controlled and configured by a set of RMCC *tcl* scripts we have developed. These scenario scripts allow us to populate the network with any desired combination of node fan-outs and produce varying time-scripted TCP and MDP traffic flows. We, additionally, control the bottleneck link capacities, link losses, and separate queueing behavior and configurations at various components in the network simulation. For post-analysis, we can easily examine the detailed results for packet loss, protocol operation, and throughput at any network point in the topology through additional post-processing tools we developed.

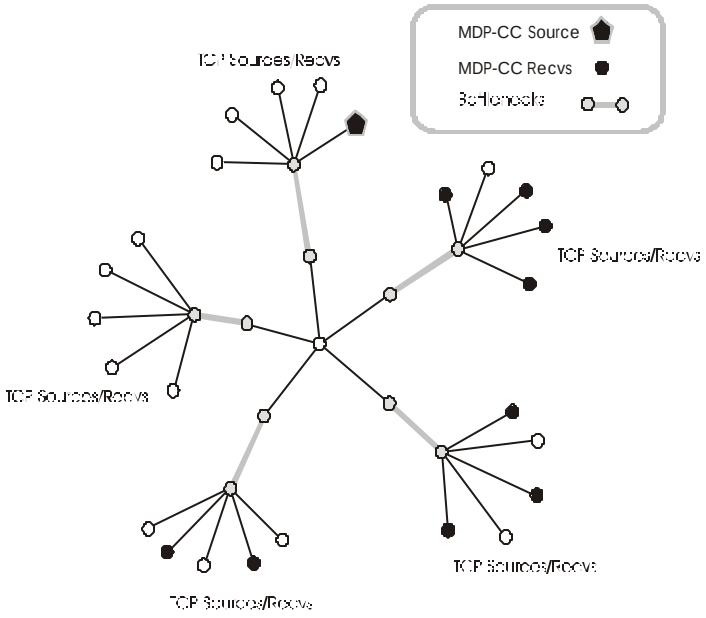

Figure 2: Multiple Bottleneck Topology Scenario

[4] UCB/LBNL/VINT Network Simulator - ns (version 2), http://www-mash.cs.berkeley.edu/ns/

Over the last year, we have performed a large set of protocol simulation studies and we here present and discuss only a snapshot of the results we have obtained.

MDP RMCC with TCP Simulation Results

One of the objectives of our initial MDP-CC work was to develop and test an end-to-end RMCC algorithm against TCP fairness. Figure 3 shows the results of a simulation to test MDP vs. TCP fairness in a single network bottleneck scenario.

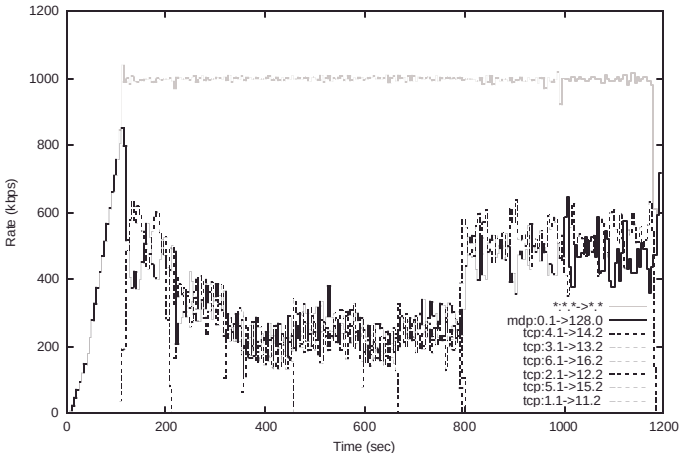

Figure 3: MDP and multiple TCP flow sharing

The basic test involves one multicast MDP flow and six dynamic TCP flows, all using different source nodes and receivers, sharing network resources over a single bottleneck topology. The bottleneck link capacity is 1Mbps and the edge network links are nominally providing 10Mbps. In previous documentation [MA99a], we provided early fairness results of a single TCP flow sharing a bottleneck queue with a single MDP flow. In Figure 3, 6 TCP sessions (e.g., emulating ftp) are used in the test with random start and stop times. All the data from Figure 3, represents the actual source rates of the various sessions, so we are viewing the actual dynamic source rate reaction of the protocols. With this simple scenario, viewing the source rate is useful and equates directly to rate fairness issues across the bottleneck. The test begins with a single MDP session under startup condition, which is ramping its source rate up towards the full 1Mbps link capacity. At approximately 100sec, the tcp-4 flow (tcp:4.1) starts and begins dynamically sharing the bottleneck bandwidth with MDP. Both flows receive around 500 Mbps, until 200 secs, at which time tcp-4 ends and tcp-3 begins. At 210 sec, a second TCP flow, tcp-6 begins, and at time 360sec a third flow, tcp-2 starts. At approximately 350-450 sec, there are 4 simultaneously TCP flows sharing bandwidth with the single MDP flow. At time 800 sec, all but one TCP session, tcp-1, has ended and there is now once again a single TCP and single MDP session sharing the bottleneck bandwidth. Note that throughout the test despite the dynamic arrival and departure of TCP sessions, all sessions, including MDP, receive a nominal fair share of the total bottleneck bandwidth available. The only protocol mechanisms at work in this example are the end-to-end congestion control algorithms of both TCP and MDP-CC.

In summary, Figure 3 shows reasonable long-term bandwidth sharing between MDP-CC multicast and TCP unicast flows under dynamic arrival and departure conditions. More elaborate testing has been performed using topologies like Figure 2, where a large number of TCP sessions occur on crosslinks with simultaneous multiple, heterogeneous bottlenecks. It is important to discover whether TCP sessions become starved on shared links or bottleneck crosslinks. The results from these early tests show good performance, but require a more detailed interpretation. Due to space limitations, those results are not presented here in favor of the more straightforward test results shown in Figure 3.

ECN Simulation Results and Loss Reduction

We are also performing analysis beyond the pure end-to-end aspects of MDP-CC algorithms and are looking into congestion notification mechanisms for RMCC performance enhancement. The MDP-CC model has been modified to recognize and use ECN bit fields in addition to measuring end-to-end packet loss at receivers. When a RED/ECN capable router marks a forwarding packet with an ECN bit, it is providing *early warning* indication of queue congestion. If pure RED queueing is running without ECN, this usually means a packet marked by ECN would have undergone early dropping from a router queue. By using ECN bit flags prior to a packet being dropped by the router, we can potentially improve loss and throughput characteristics of protocols while maintaining dynamic fair sharing. To demonstrate this, Figure 4 shows the loss results of a two fair sharing test across a single bottleneck scenario.

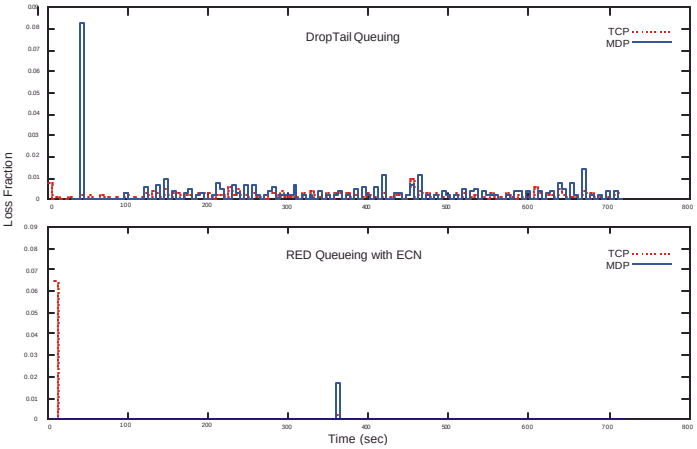

Figure 4: Packet Loss Comparison (FIFO/DT and RED/ECN)

In the top figure, Figure 4a, MDP and TCP ran over a network of FIFO droptail queues using pure end-to-end sensing and reaction algorithms. The reader should note that nominal loss is consistently occurring for both transport protocols. The consequence is increased retransmission processing and overall delay in the reliable delivery of data. In the bottom figure, Figure 4b, results are shown with RED/ECN queueing for both TCP-SACK and MDP-CC with ECN awareness. Loss statistics due to queue overflow are nominally zero with a few exceptions (note: in Figure 4 protocol startup causes temporary loss). We are using

random start times for the two sessions in Figure 4a-b and have run many trials to verify consistent loss behavior as observed here. The bandwidth sharing fairness results (not shown) between trials 4a and 4b are nearly equivalent (i.e. MDP-CC and TCP share the bandwidth roughly equally in both cases. The point of the graphs is to illustrate the resultant packet loss results.

In addition to decreasing end-to-end loss and improving efficiency, the use of congestion notification may be of further benefit for wireless protocol application. The use of RED/ECN provides a method in effect to partially separate packet loss reaction --which may occur frequently on wireless IP links-- from queue congestion reaction. If robust enough, the ECN bit signaling can perhaps provide a mechanism for wireless transport improvements and further research. TCP-SACK as tested in Figure 4 may also provide additional wireless benefits with its more selective retransmission behavior. This selective retransmission behavior is already a built-in behavior of MDP-CC. Early RED/ECN results are encouraging us to further study this area in future work applied to dynamic wireless networks.

CONCLUSIONS

We have provided an overview of RMCC design issues and discussed specific work done on congestion control extensions to MDP, MDP-CC. Our design objectives were to develop an RMCC protocol extension that worked end-to-end, met reasonable TCP fairness criteria, and also maintained the scalable properties of NACK-oriented design. These scalable properties involve NACK suppression, control aggregation, and erasure-based packet repairing components. We described our simulation efforts and showed an example of dynamic bottleneck sharing fairness between MDP-CC and dynamic TCP sessions. Additional results (not presented due to space limitations) are generally encouraging and consistently demonstrate a degree of reasonable fairness and robustness for coexisting MDP-CC and TCP traffic sessions under a varied set of conditions. We also demonstrated through simulation that the addition of router congestion notification (e.g., RED/ECN) can improve end-to-end application data throughput and protocol efficiency by reducing packet loss due to congestion. This improvement was achieved while maintaining MDP-CC and TCP fair sharing behavior.

REFERENCES

[RFC793] Transmission Control Protocol (TCP), DARPA Internet Protocol Specification, September 1981.

[Jacobson88] V. Jacobson, *"Congestion Control and Avoidance"*, Proc. of SIGCOMM 1988, pp. 314-328.

[Stevens97] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", Internet RFC 2001, January 1997.

[MCK96] Macker, J., M.S. Corson, E. Klinker, *"Reliable Multicast Data Delivery for Military Internetworking"*, Proc. IEEE MILCOM 96, Oct 1996.

[MRBP98] A. Mankin, A. Romanow, S. Bradner, V. Paxson, "IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols", RFC 2357, June 1998.

[FF99] Floyd, S., and Fall, K., *"Promoting the Use of End-to-End Congestion Control in the Internet"*, IEEE/ACM Transactions on Networking, August 1999.

[MC97] J. Macker, M.S. Corson, *"Controlled Link Sharing and Reliable Multicast for Asymmetric Networks"*, Pacific Telecommunications Conference, Jan 1997.

[MP98] J. Macker, V. Park, *"Internetwork Traffic Management Enhancements: Evaluation and Experimental Results"*, in Proc. IEEE MILCOM 98, Vol. 2, pp. 609-613, Oct 1998.

[Macker97a] J. Macker, "Reliable Multicast Transport and Integrated Erasure-based Forward Error Correction", Proc. IEEE MILCOM 97, Oct 1997.

[Macker97b] J. Macker, "Integrated Erasure-Based Coding for Reliable Multicast Transmission", IRTF Meeting presentation, March 1997.

[RMTWG] http://www.ietf.org/html.charters/rmt-charter.html

[ABFHM00] Adamson, Borman, Floyd, Handley, Macker, "*NACK-Oriented Reliable Multicast (NORM) Protocol Building Blocks,"* IETF draft-rmt-norm-bb-00.txt, work in progress.

[MA99a] J. Macker, R. B. Adamson, *"The Multicast Dissemination Protocol Toolkit"*, Proc. IEEE MILCOM 99, Oct 99.

[MA99b] J. Macker, B. Adamson. "*The Multicast Dissemination Protocol (MDP)*," draft-macker-rmt-mdp-00.txt, Internet Draft, work in progress, Oct 99.

[PFTK98] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, *"Modeling TCP Throughput: A Simple Model and its Empirical Validation"*, Univ of Mass, Technical Report CMPCSI TR 98-008.

[WC98] B. Whetton, J. Conlan. *"A Rate Based Congestion Control Scheme for Reliable Multicast"*, Technical White Paper, Oct 1998.

[DO98] D. DeLucia, K. Obraczka, *"Congestion Control Performance of a Reliable Multicast Protocol.,"*. Proc. of IEEE ICNP'98, August 1998.

[PTK93] S. Pingali, D. Towsley, J. Kurose. *"A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols"*, In Proc. INFOCOM, San Francisco, CA, October 1993.

[LG96] B.N. Levine, J.J. Garcia-Luna-Aceves. "A Comparison of Known Classes of Reliable Multicast Protocols", *Proc. International Conference on Network Protocols (ICNP-96)*, Columbus, Ohio, Oct 29--Nov 1, 1996