Coding for Improved Throughput Performance in Network Switches

ABSTRACT

With the increasing demand for network bandwidth, network switches (and routers) face the challenge of serving growing data rates. Currently the most viable way to scale switching rates is by parallelizing the writing and reading of packets between multiple memory units (MUs) in the switch fabric. However, this introduces the problem of *memory contention*, whereby multiple requested packets need to access the same bandwidth-limited MUs. Our ability to avoid such contention in the write stage is limited, as the reading schedule of packets is not known upon arrival of the packets to the switch. Thus, efficient packet placement and read policies are required, such that memory contention is mitigated.

EXISTING SYSTEM

In Existing System, for greater flexibility in the read process, coded switches introduce *redundancy* to the packet-write path. This is done by calculating additional coded chunks from an incoming packet, and writing them along with the original packet chunks to MUs in the switch memory. A coding scheme takes an input of k packet chunks and encodes them into a codeword of n chunks ($k \leq n$), where the redundant $n - k$ chunks are aimed at providing improved read flexibility. Thanks to the redundancy, only a subset of the coded chunks is required for reconstructing the original (uncoded) packet. Thus, packets may be read even when only a part of their chunks is available to read without contention. One natural coding approach is to use $[n, k]$ *maximum distance separable* (MDS) codes, which have the attractive property that *any* k chunks taken from the n code chunks can be used for the recovery of the original k packet chunks. Although MDS codes provide the maximum flexibility, we show in our results that good switching performance can be obtained even with much weaker (and lower cost) codes, such as binary cyclic codes.

DIS ADVANTAGES

- It cannot be used for the recovery of the original packet chunks.

- The problem of obtaining maximum throughput for a set of requested packets is a hard problem.

PROPOSED SYSTEM

In Proposed System, network switches, and in particular, how to design them to maximize the throughput advantages over standard uncoded switches. Toward that objective, the paper contributes a variety of algorithmic and analytical tools to improve and evaluate the throughput performance. The most interesting finding of this paper is that the placement of packets in the switch memory is the key to both high performance and algorithmic efficiency. One particular placement policy we call “design placement” is shown to enjoy the best combination of throughput performance and implementation feasibility.

ADVANTAGES

- It achieves higher utilization of the memory units.
- To reduce the amount of contention in packet reading.

SYSTEM REQUIREMENTS

H/W System Configuration:-

Processor	-	Pentium –III
RAM	-	256 MB (min)
Hard Disk	-	20 GB
Key Board	-	Standard Windows Keyboard
Mouse	-	Two or Three Button Mouse
Monitor	-	SVGA

S/W System Configuration:-

Operating System	:	Windows95/98/2000/XP
Application Server	:	Tomcat5.0/6.X
Front End	:	HTML, Jsp
Scripts	:	JavaScript.
Server side Script	:	Java Server Pages.
Database	:	MySQL 5.0
Database Connectivity	:	JDBC