Digital Communication II 2004 – Paper 8 Question 3 (IAP/JAC)

(*a*) A modern router consists of a line cards interconnected by a space division switch fabric (as opposed to a shared bus). There will also be a CPU/memory card connected to the switch fabric, but this is only involved with control plane operations (running routing daemons, calculating the forwarding information base) and dealing with non-standard packets (e.g. IP options).

Line cards contain link-specific physical and MAC layer parts and then components for IP header processing (header checksum check, TTL decrement, checksum recalculation), forwarding table lookup, and then buffering prior to transmission across the switch fabric. The egress path may contain further buffering before packets are transmitted by the MAC/phy.

(*b*) (*i*) The variable length prefixes in the forwarding table can be represented as a binary trie (most significant bit at the root of the tree). The leaves of the trie would contain the next hop forwarding information. This is a compact and simple to maintain representation. However, lookups are potentially slow as they require walking the tree, requiring a memory read at every level. /24 prefixes are quite common in core router tables, and hence such lookup would require 24 reads. Since the table is relatively small, it might be possible to use SRAM.

(*ii*) The most significant 24 bits of incoming addresses could be used directly as an index into a large lookup table. Entries in the table would either contain the next hop information, or a pointer to another 256 entry lookup table that will then be indexed with the bottom 8 bits of the address to yield the next hop information.

The large lookup table uses significant amounts of memory, but DRAM is cheap. There are lots of duplicated entries which must be maintained when a route changes. However, lookups have good performance, requiring just one or two DRAM reads.

(*iii*) One possible strategy would be to employ multiple hash tables, one for each length of prefix (e.g. 1-32). The appropriate number of bits of the address would concurrently be looked up in each table. A priority encoder would be used to select the longest matching prefix in case of hits in multiple hash tables.

(*c*) Output buffered switches are easy to model from a queueing theory point of view, but are generally not practical to build as they would require a switch fabric capable of simultaneously transferring packets from multiple input ports to the same output concurrently. Hence for an N port switch the bandwidth into a single output port must be N times the link rate (L).

Input buffered switch fabrics suffer from head of line blocking. Each output port

can only accept packets from a single input at a time. Input ports implement a simple FIFO queue of packets, hence no reordering is possible. Thus, if the head of the fifo at more than one input port is destined to the same output, the arbiter can only grant one transfer, and hence the other input ports will stall. This head of line blocking significantly reduces the usable throughput of the fabric. With a random traffic mix, loads above 65% will cause runaway queueing.

A virtual output buffered switch fabric is an improvement on the input buffered design. Each input port maintains N fifos, one per destination output port. Each input port tells the arbiter which output ports it has packets for. The arbiter than performs a bipartite match (ideally a maximal weight match) and hence ensures that a non conflicting set of transfers are initiated.