# Deadlock- and Livelock-Free Packet Switching Networks

*Sam Toueg*†

IBM T.J. Watson Research Center
P.O.Box 218
Yorktown Heights, NY 10598

*ABSTRACT*

A controller for a packet switching network is an algorithm to control the flow of packets through the network. A local controller is a controller executed independently by each node in the network, using only local information available to these nodes. A controller is deadlock- and livelock-free if it guarantees that every packet in the network reaches its destination within a finite amount of time. We present a local controller which is proved to be deadlock- and livelock-free.

## 1. Introduction

### 1.1. Basic Definitions

A *packet switching network* is a directed graph $G=(V, E)$; the vertices $V$ represent processors, and the edges $E$ represent communication links. We assume messages, called *packets,* are to be passed between processors. Each network has an associated constant $b$, the number of *buffers* at each vertex; a buffer can hold exactly one packet. Associated with each packet is an acyclic *route* $v_1, v_2, \ldots, v_q$, which is a path in $G$. Vertex $v_1$ is the *source,* and $v_q$ is the *destination* vertex for the packet. We assume a *fixed routing procedure* [KL], where a packet's route is determined at the source node. We may also assume that the route of a packet is included as part of the message in the packet, although in practice the packet could hold only the source and destination, with each processor in the network responsible for deducing the next vertex to which the packet is to be passed. Also associated with a network $G$ is the constant $k$, the length of the longest route taken by a packet in $G$. If we want to state explicitly the two constants associated with a network $G$ we write that $G$ is a *(b, k)-network.* Note that $k$ need not be the length of the longest path in the network; we may never wish to send messages between distant nodes.

The *moves* made by the network are of three types.

1. *Generation.* A vertex $v$ accepts, and places in an empty buffer, a packet $p$ created by a process $P$ residing in $v$.
2. *Passing.* A vertex $v$ transfers a packet in one of its buffers to an empty buffer of vertex $w$, where $v \rightarrow w$ is an edge, and the route for the packet has $w$ following $v$. The buffer of $v$ holding the packet becomes empty.
3. *Consumption.* A packet in a buffer of $v$, such that the destination for the packet is $v$, is removed from that buffer and the buffer is made empty.

### 1.2. Controllers, Deadlocks and Livelocks

A *controller* for a network is an algorithm that permits or forbids various moves in the network. One of the key problems in packet switching is preventing *deadlocks,* which are situations in which one or more packets can never reach their destination no matter what sequence of moves is subsequently performed. For example, in the network of Fig. 1, if all physically possible moves are permitted by the controller, $v_1$ generates $b$ packets with destination $v_2$, $v_2$ does the same with destination $v_3$, and $v_3$ does the same with destination $v_1$, then all buffers of all vertices will be full, no consumption moves can take

† This work was supported in part by the National Science Foundation under Grant GK-42048, and in part by the U.S. Army Research Office, Durham, NC under Grant DAAG29-75-0192, while the author was at Princeton University, Princeton, New Jersey.

place without a pass move, and no generation can take place. It is not hard to see that the network is deadlocked.

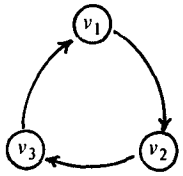


**Fig. 1.** A network exhibiting deadlock with a trivial controller.

Several *deadlock-free controllers,* i.e. controllers that prevents deadlocks, are known to date [G, RH, TU], but none of them prevents another kind of failure to be avoided in packet switching networks: *livelock.* Livelock is a situation in which unfair scheduling of packets dynamically prevents one or more packets to reach their destination. For example, the following simple controller, the *Forward-Count* (or *FC*) *controller,* was proved to prevent deadlocks in any network [TU]. With FC a packet $p$ may be passed to or generated into a node $v$‡ if and only if the number of free buffers in $v$ is greater than the distance from $v$ to the destination of $p$ along the packet's route. A simple network, illustrated in Fig. 2, shows that FC does not prevent livelocks.

The network consists of two routes, $r_1$:$(v_1, v_2, \ldots, v_{10})$, and $r_2$:$(w_1, v_2, w_3)$; every node has ten buffers. With FC, a packet generated in $v_1$ (a "white packet") may be passed to $v_2$ if and only if $v_2$ has at least 9 free buffers, but for a packet generated in $w_1$ (a "black packet") to be accepted by $v_2$ it is enough for $v_2$ to have two (or more) free buffers. Under heavy input load from the source node $w_1$, this imbalance soon causes the following livelock. Black packets fill up most of $v_2$'s buffers; the departure of any black packet $p$ from $v_2$ frees a buffer, but the number of free buffers in $v_2$ is still not large enough (i.e., less than 9) for a white packet, waiting in $v_1$, to enter $v_2$;

‡ From now on we may use "accepted by a node" instead of "passed to or generated into a node"

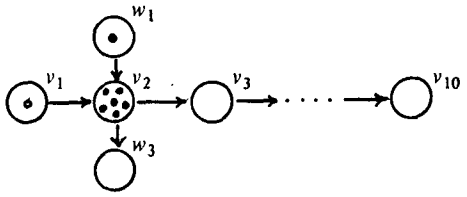


**Fig. 2.** Livelock occuring with the FC controller.

soon, a black packet generated in $w_1$ enters $v_2$ and replaces the departed packet $p$. This dynamic process repeats, and, as long as the flow of packets in $r_1$ is not reduced, the white packet in $v_1$ can never reach its destination (the white packet is *livelocked*).

A *livelock-free* (or *LF*) *uniform controller* is one that prevents livelocks in any network. We are looking for a *(b, k)-deadlock and livelock-free* (or *DLF*) *uniform controller,* one that prevents both deadlocks and livelocks in any $(b, k)$-network.† With a DLF controller every packet $p$ reaches its destination within a finite amount of time from the moment of its creation.‡

### 1.3. Local Controllers

Whereas in general a controller can examine the state of the entire network, we do not consider that this is a reasonable assumption. There has to be at least one vertex at which the controller resides, and this vertex would have to be connected directly to every other vertex, which requires an arbitrary number of connections to one processor or packets informing the controller of local conditions

† We shall omit the parameters $(b, k)$ whenever they are understood: "uniform controller" stands for "$(b, k)$ uniform controller"

‡ It should be clear that the creation of a packet by a process in a node precedes the generation of the packet in this node.

would have to be passed around the network, and this information would itself alter the state of the network (and generate too much message traffic).

Thus we shall restrict ourselves to *local controllers,* where each processor decides on the legality of accepting a packet and the decision is made according to *local information* alone. To this purpose, we shall consider the following local information defining node and packet states in a $(b, k)$-network $G$. We characterize the *local state* of a node $v$ of $G$ by the following parameters,

$\mathbf{j}=<j_0, j_1, \ldots, j_k>$, where $j_i$, $(0 \leqslant i \leqslant k)$, is the number of packets in $v$ whose destination distance from $v$ is $i$, and

$\mathbf{t}=<t_0, t_1, \ldots, t_k>$, where $t_i$, $(0 \leqslant i \leqslant k)$, is the earliest (i.e., the smallest) creation time among the packets waiting to be accepted by $v$ whose distance from $v$ to their destination is less or equal to $i$. If there are no such packets it is convenient to set $t_i=\infty$.*

Similarly, any packet $p$ asking access to a node $v$ has a *packet state relative to* $v$ characterized by the following parameters,

$j$, the distance from the node $v$ to the destination of $p$, and

$t$, the creation time of the packet $p$.

Note that, with our assumption of fixed routing policy, and if the creation time of a packet is stored in the packet, all the local information listed above is readily available to any node $v$ when it is considering whether to accept a packet $p$.

In our formalism, we shall define an $(\alpha, \beta)$ controller $S$ to be a set of $(\alpha, \beta)$ tuples where $\alpha$ denotes a node state and $\beta$ denotes a packet state. $(\alpha_0, \beta_0)$ is in $S$ if and only if it is permissible for an $\alpha_0$-state vertex to accept a $\beta_0$-state packet.

* Clearly, $t_c$, the creation time of a packet $p$, and $t_e$, the time elapsed from the creation of $p$, are related at any time $t$ by $t_e=t-t_c$, i.e. the earliest creation time corresponds to the greatest elapsed time.

## 2. Local DF and DLF Uniform Controllers

### 2.1. The Forward-State Controller

Several DF uniform local controllers were investigated in [TU]; one of them is the *Forward-State* (or *FS*) *controller.* For $b>k$, $FS(b, k)$, or just FS when $(b, k)$ is understood, is the set

$$\{(\mathbf{j}, j) \mid \text{for all } i_k\ 0 \leqslant i \leqslant j,\ i < b-\sum_{r=i} j_r, \text{ and } 0 \leqslant j \leqslant k \text{ and } 0 \leqslant \sum_{r=0} j_r \leqslant b-1\}$$

In the above, we assume $\mathbf{j}=<j_0, j_1, \ldots, j_k>$.

**Theorem 1** [TU]. FS is a DF uniform controller.

The DLF controller described in the next section is based on some properties of FS. To state them we must first introduce the concept of reachability of node states with respect to $(\alpha, \beta)$ DF uniform controllers. Let $S$ be such a controller and let $(\alpha_0, \beta_0) \in S$. Suppose the acceptance of a $\beta_0$-state packet into an $\alpha_0$-state node results in an $\alpha_0'$ node state, we denote this state change by

$$\alpha_0 \overset{\beta_0 |}{\underset{S}{\vdash}} \alpha_0'$$

Similarly, if a $\beta_0$-state packet leaves an $\alpha_0$-state node and this results in a new state $\alpha_0'$, then we denote this change by

$$\alpha_0 \overset{\beta_0 |}{\underset{S}{\vdash}} \alpha_0'$$

In both cases, if we are only interested in the state transition we just write $\alpha_0 \underset{S}{\vdash} \alpha_0'$. $\underset{S}{\vdash}$ is a "state transition" relation in the set of states. The transitive closure of the relation $\underset{S}{\vdash}$ is denoted by $\overset{*}{\underset{S}{\vdash}}$.

Let $S$ be an $(\alpha, \beta)$ DF uniform controller and $\alpha_0$ and $\alpha_0'$ be node states. We say that $\alpha$ is *reachable from* $\alpha_0$ *with respect to* $S$ if and only if $\alpha_0 \overset{*}{\underset{S}{\vdash}} \alpha_0'$. If $\alpha_0$ is the "empty

node" state then we simply say that $\alpha$ is *reachable with respect to S* and we denote this fact by $\vdash_S^* \alpha$ (the *empty node state* $\alpha_0$ is the state description of a node without any packet stored in its buffers or any packet waiting to be accepted by the node; with our parameters, $\alpha_0$ is characterized by the following values $\mathbf{j}=<0, 0, \ldots, 0>$, and $\mathbf{t}=<\infty, \infty, \ldots, \infty>$).

**Lemma 1.** If $\vdash_{FS}^* \mathbf{j}=<j_0, j_1, \ldots, j_k>$, then for all $i$, $0 \leqslant i \leqslant k$, we have $b-\sum_{r=i}^{k} j_r \geqslant i$.

*Proof.* Suppose $\vdash_{FS}^* \mathbf{j}=<j_0, j_1, \ldots, j_k>$, then $\vdash_{FS}^q \mathbf{j}$ for some $q \geqslant 0$. By induction on $q$ we prove that if $\vdash_{FS}^q \mathbf{j}$ then for all $i$, $0 \leqslant i \leqslant k$, $b-\sum_{r=i}^{k} j_r \geqslant i$. If $\vdash_{FS}^0 \mathbf{j}$, then $\mathbf{j}$ is the empty-node state. So $j_0=j_1=\cdots=j_k=0$, and since $b>k$ the induction hypothesis holds. Suppose that the induction hypothesis holds for $q=t-1$, we show that it must also hold for $q=t$. If $\vdash_{FS}^t \mathbf{j}$ then $\vdash_{FS}^{t-1} \mathbf{j}^0$ and $\mathbf{j}^0 \vdash_{FS} \mathbf{j}$ for some node state $\mathbf{j}^0=<j_0^0, j_1^0, \cdots, j_k^0>$. By induction hypothesis, for all $i$, $0 \leqslant i \leqslant k$, we have $b-\sum_{r=i}^{k} j_r^0 \geqslant i$. If $\mathbf{j}^0 \vdash_{FS}^{\beta\downarrow} \mathbf{j}$ then

$$\begin{cases} j_i=j_i^0 & \text{for } i \neq \beta \\ j_i=j_i^0-1 & \text{for } i=\beta \end{cases}$$

and the induction hypothesis will obviously hold for the state $\mathbf{j}$. If $\mathbf{j}^0 \vdash_{FS}^{\beta\uparrow} \mathbf{j}$ then $(\mathbf{j}^0, \beta)$ is in FS. Therefore, for all $i$, $0 \leqslant i \leqslant \beta$, $b-\sum_{r=i}^{k} j_r^0 > i$, and we have

$$\begin{cases} j_i=j_i^0 & \text{for } i \neq \beta \\ j_i=j_i^0+1 & \text{for } i=\beta \end{cases}$$

Then, for all $i$, $0 \leqslant i \leqslant \beta$, $b-\sum_{r=i}^{k} j_r \geqslant i$. This last result combined with the induction hypothesis shows that for all $i$, $0 \leqslant i \leqslant k$, $b-\sum_{r=i}^{k} j_r \geqslant i$. □

**Lemma 2.** If $\vdash_{FS}^* \mathbf{j}=<j_0, j_1, \ldots, j_k>$ and for some $d$, $0 \leqslant d \leqslant k$, we have $b-\sum_{r=d}^{k} j_r=d$, then $j_d \geqslant 1$.

*Proof.* If $d=k$ then $b-j_k=k$, so $j_k=b-k \geqslant 1$. If $0 \leqslant d<k$ then $b-\sum_{r=d+1}^{k} j_r=b-\sum_{r=d}^{k} j_r+j_d$, therefore $b-\sum_{r=d+1}^{k} j_r=d+j_d$. By Lemma 1 we have $b-\sum_{r=d+1}^{k} j_r \geqslant d+1$, so $d+j_d \geqslant d+1$, and $j_d \geqslant 1$. □

Let $\mathbf{j}=<j_0, j_1, \ldots, j_k>$ be a reachable state with respect to FS. We define $d(\mathbf{j})$, the *distance threshold* of the state $\mathbf{j}$, as follows,

$$d(\mathbf{j})=\min\{\, i \mid b-\sum_{r=i}^{k} j_r=i \text{ or } i=k+1 \,\}$$

**Lemma 3.** If $\vdash_{FS}^* \mathbf{j}$ and $d(\mathbf{j})$ is the corresponding distance threshold, then $(\mathbf{j}, d) \in FS$ if and only if $0 \leqslant d < d(\mathbf{j})$.

*Proof.* Let $\vdash_{FS}^* \mathbf{j}$ and let $d^*=d(\mathbf{j})$ be the distance threshold of $\mathbf{j}$. If $d^*=k+1$ then for all $i$, $0 \leqslant i \leqslant k$, we have $b-\sum_{r=i}^{k} j_r > i$. Directly from the definition of FS we can now verify that $(\mathbf{j}, d) \in$ FS if and only if $0 \leqslant d < d^*=k+1$. If $d^*<k+1$ then $b-\sum_{r=d^*}^{k} j_r=d^*$, and for all $i$, $0 \leqslant i < d^*$, we have $b-\sum_{r=i}^{k} j_r > i$. It is clear now that if $0 \leqslant d < d^*$ then $(\mathbf{j}, d) \in$ FS, and if $d \geqslant d^*$ then $(\mathbf{j}, d)$ is not in FS. □

**Lemma 4.** Let $G=(V, E)$ be a $(b, k)$-network, and $v \in V$ be a node with the state $\mathbf{j}$ reached with respect to FS. Let $d^*=d(\mathbf{j})$ be the corresponding distance threshold. If $d^* \neq k+1$ then

1. $j_{d^*} \geqslant 1$, i.e. there is at least one packet $p^*$ in $v$ whose distance from $v$ to its destination is $d^*$.
2. If the packet $p^*$ ever leaves the node $v$ the resulting state $\mathbf{j}_0$ has a distance threshold $d^0$ such that $d^0 > d^*$.

*Proof.*

1. If $d^* \neq k+1$ then $b-\sum_{r=d^*}^{k} j_r = d^*$, and, by Lemma 2, we have $j_{d^*} \geqslant 1$.
2. Let $\mathbf{j}^1$ and $\mathbf{j}^0$ be the states of the node $v$ immediately before and after the packet $p^*$ leaves the node $v$. Since $\vdash_{FS}^{*} \mathbf{j}^1$ then, by Lemma 1, for all $i$, $0 \leqslant i \leqslant k$, we have $b-\sum_{r=i}^{k} j_r^1 \geqslant i$. After the departure of $p^*$ we have $j_{d^*}^0 = j_{d^*}^1 - 1$, and all the other components of $\mathbf{j}^0$ and $\mathbf{j}^1$ are equal. Therefore for all $i$, $0 \leqslant i \leqslant d^* \leqslant k$, we have $b-\sum_{r=i}^{k} j_r^0 > i$, and, by definition, $(\mathbf{j}^0, d^*)$ must be in FS.‡ By Lemma 3, the distance threshold $d^0=d(\mathbf{j}^0)$ is such that $0 \leqslant d^* < d^0$. □

We are now ready to state and prove the main result of this paper.

### 2.2. The Forward-State Elapsed-Time Controller

We define the *Forward-State Elapsed-Time controller* as follows. For $b > k$, let FSET$(b, k)$, or just FSET when $(b, k)$ is understood, be the following set,

$$\{[(\mathbf{j}, \mathbf{t}), (j, t)] \mid 0 \leqslant j < d^*=d(\mathbf{j}) \quad \text{and} \quad 0 \leqslant t \leqslant t_{d^*-1}\}$$

where $\mathbf{j}$, $\mathbf{t}$, $j$ and $t$ are the parameters we defined in section 1.3. With FSET, an arriving packet is accepted by a node $v$ if and only if this is permissible with respect to the controller FS and there are no waiting packets† with a greater elapsed time which may also be accepted by the node according to FS.

**Theorem 2.** FSET is a DLF uniform controller.

*Proof.* FSET is a deadlock-free uniform controller, the proof is similar to the one given in [TU] for FS. We must show that FSET is a livelock-free uniform controller, i.e. any packet reaches its destination after a finite amount of time in any given network. Suppose $G$ is a $(b, k)$-network where a livelock situation is reachable with FSET; let $L$ be the maximal set of livelocked packets in this livelock. There is a time $t_0$ when

1. all the packets in $L$ have reached the node in their route where they remain "blocked forever", and
2. all the packets in $L$ have a greater elapsed time than any other packet in the network.

We consider the network behavior from the time $t_0$ on. Let $p_1 \in L$ be a livelocked packet in a node $v_1$; $p_1$ is waiting to be passed to a node $v_2$.* Let $d_1$ be the distance from $v_1$ to the destination of $p_1$, and let $t_1$ be the creation time of $p_1$. Then, with respect to $v_2$, the local state of $p_1$ is $(d_1-1, t_1)$. Let $d_2$ be the maximal distance threshold reached by the states of node $v_2$ (from the time $t_0$ on). The state of $v_2$, at the time $t_2$ the maximal distance threshold $d_2$ is first achieved, will be denoted by $(\mathbf{j}_2, \mathbf{t}_2=<\tau_1, \tau_2, \ldots, \tau_k>)$, by

‡ This is essentially a proof that any packet which exits a node can immediately reenter this node according to the FS controller. This property is essential in our proofs, and it holds only for the state controllers. With the BC or FC controller, it is easy to provide an example of a reachable state in a node where a packet that leaves this node is not allowed to reenter the node.

† Packets whose request to be passed or generated into $v$ was previously denied and has not yet been granted.

* Another case to be considered is the following, $p_1 \in L$ was created in a node $v_1$ and is waiting to be *generated* in $v_1$. The proof of this case is similar to the one presented here.

our definitions $d_2 = d(j_2)$. Suppose $d_1 - 1 < d_2$. We will show that this leads to a contradiction. Let $W$ be the set of waiting packets, at the time $t_2$, whose access to $v_2$ is permissible with respect to *FS* (i.e., whose distance from $v_2$ to destination is less than the threshold $d_2$); $W$ contains $p_1$. We denote by $p_1^*$ the packet in $W$ with the earliest creation time $t_1^*$. We must have $\tau_{d_2-1} = t_1^* \leqslant t_1$, therefore $p_1^*$ is in $L$. It should be clear that from the time $t_2$ on, as long as $p_1^*$ is waiting to be accepted by $v_2$, no other packet can be passed or generated into $v_2$ according to FSET. So, the distance threshold $d_2$ does not decrease, and when $p_1^*$ is eventually retransmitted† to the node $v_2$ the packet is accepted according to FSET. But $p_1^*$ is in $L$ and a contradiction results; therefore we must have $d_2 < d_1$. Since $d_1 > d_2$ then $d_2 \neq k+1$ and, by Lemma 4, there is a packet $p_2$ in $v_2$ whose distance from $v_2$ to destination is $d_2$, and if $p_2$ ever leaves the node $v_2$ then the resulting distance threshold will be greater than $d_2$. This is impossible by the maximality of $d_2$, therefore $p_2$ never leaves the node $v_2$: $p_2$ must be a livelocked packet (i.e., $p_2 \in L$). A repetition of the argument used before shows that there is a sequence of livelocked packets $p_1, p_2, \cdots, p_l, \cdots$ whose distances to destination $d_1 > d_2 > \cdots > d_l > \cdots$ are monotonically decreasing. Then we can prove the existence of a livelocked packet whose distance to destination is zero; but such a packet would be consumed after a finite amount of time. □

We may note that the addition of the creation time parameter and the use of the same strategy with the Forward-Count controller instead of the FS controller, do not yield a DLF uniform controller as could be expected. Of the four local controllers introduced in [TU], the Backward-State controller BS is the only one besides FS whose modification results in a DLF uniform controller.

† In packet switching networks waiting packets are periodically retransmitted until they are accepted by their target node.

**References**


[G] K. D. Gunther, "Prevention of Buffer Deadlocks in Packet Switching Networks," IFIP-IIASA workshop on data communications, Ladenberg, Austria, g.15-g.19, Sept., 1975.

[GD] M. G. Gouda, "Protocol machines," Ph. D. thesis, Dept. of C. S., Univ. of Waterloo, Waterloo, Canada, 1977.

[GHKP] A. Giessler, J. Haenle, A Koenig, and E. Pade, "Free Buffer Allocation - an Investigation by Simulation," *Computer Networks,* 2 (1978), pp. 191-208.

[KL] L. Kleinrock, *Queueing Systems* Vol. II: *Computer Applications,* Wiley, N. Y., 1976.

[RH] E. Raubold and J. Haenle, "A Method of Deadlock-Free Resource Allocation and Flow Control in Packet Networks," *Proc. ICCC 76,* Toronto, Ont., Canada, Aug., 1976, pp. 483-487.

[TU] S. Toueg and J. D. Ullman, "Deadlock-Free Packet Switching Networks," *Proc. ACM Symposium on the Theory of Computing,* Atlanta, Georgia, May., 1979, pp. 89-98.