Deadlock, Livelock, and Starvation

Outline

- **Situations preventing packet delivery**
- **Theory of Deadlock Avoidance**
  - Channel Dependency
  - Necessary and sufficient condition
  - Relaxing this constraint
  - Extended Channel Dependency Graph
- **Deadlock in MINs**

Situations preventing packet delivery

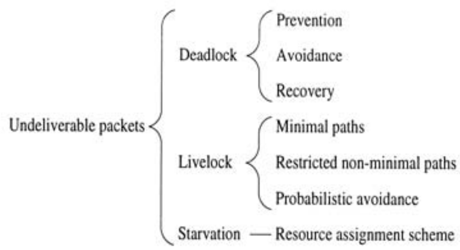

Figure 3.1: A classification of the situations that may prevent packet delivery

Theory of Deadlock Avoidance

- **Basic definitions**
  - **routing function (set of output channels based on the current and destination nodes)**
  - **selection function (one free channel is selected out of many, if available)**
  - **configuration (assignment of a set of packets or flits to each queue) based on a *reachable* or *routable* assignment**
    - **legal**
    - **illegal**
  - **connected routing function (able to establish a path between each pair of nodes)**

Deadlock Configuration

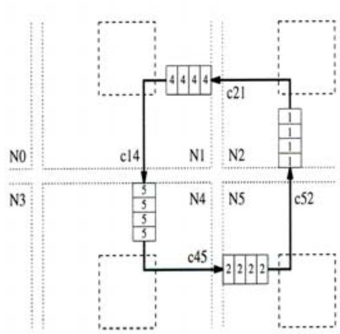

Figure 3.2: Deadlocked configuration for R

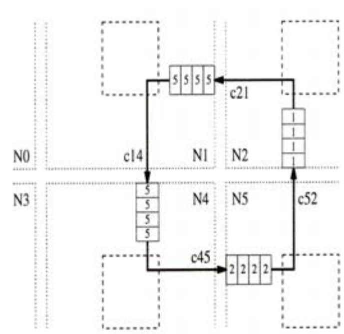

Figure 3.3: Illegal configuration for R

Avoiding Deadlock

- **Identification of *channel dependency* under a routing function R**
- **Necessary and Sufficient Condition**
  - **Breaking the dependency by providing some restriction on R**

Breaking Channel Dependency with Virtual Channels on a Ring

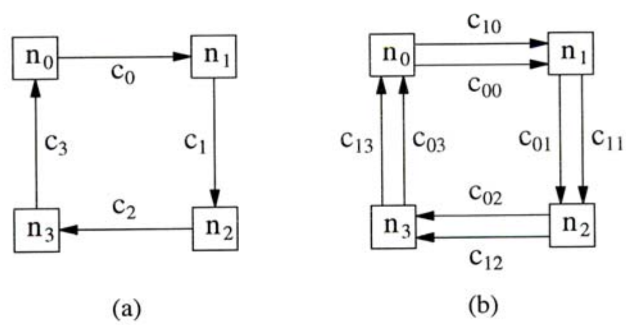

Figure 3.4: Networks for the example

Can we Avoid Deadlock in the Presence of Cyclic Dependency?

- **No until 1993**
- **All networks were built with the notion of breaking cyclic dependency**
- **Prof. Duato provided a new framework**
  - **concept of escape channel**
  - **routing subfunction corresponding to escape channels should not have cyclic dependency**
  - **has led to flexible routing schemes**
  - **led to simpler adaptive routing schemes**

Deadlock Avoidance in the Presence of Cyclic Dependency

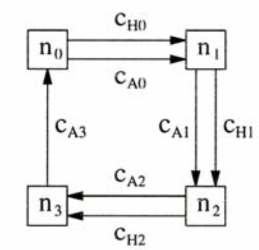

Figure 3.5: Network for the example

Use either C_{Ai} (j≠i) or C_{Hi} j>i

Extended Channel Dependency Graph

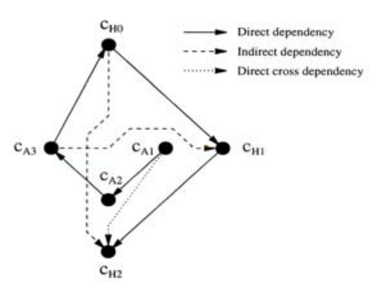

Figure 3.8: Extended channel dependency graph for R_1 using wormhole switching

R1: use C_{Ai} if j<i or C_{Hi} if j>i

Developing Adaptive Routing Algorithms

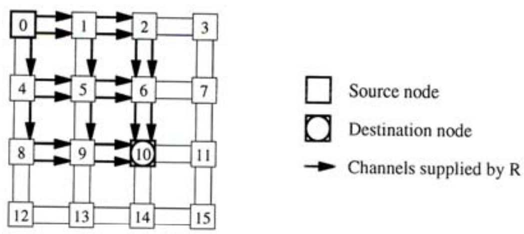

Figure 3.9: Routing example for R

Deadlock in MINs

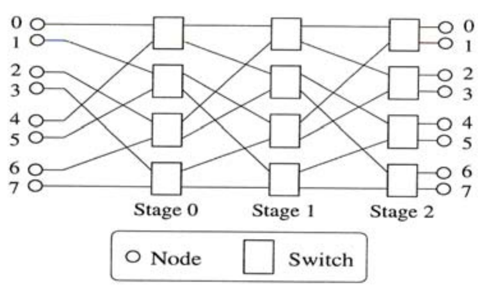

Figure 3.13: A multistage interconnection network

Deadlock Recovery

- Recently has generated some interest
- Deadlock Avoidance restricts some use of channels/buffers
- Can we trade-off between better resource utilization and deadlock recovery?
- Reserve minimal resources for recovery
- Should limit which packets in deadlock are eligible for recovery

Deadlock Recovery (cont.)

- Recovery Process
  - Progressive
  - Regressive
- Deadlock detection vs. blocking
- Heuristics for deadlock detection
  - time-outs (value dependent on message length)
  - may not detect deadlock for a long time or
  - may "detect" deadlocks that do not exist
- not applied to any real system yet

Livelock

- Occurs if a packet circulates forever in the network
- Limiting misrouting avoids livelock
- Minimal routing algorithms allow no misrouting
- Misrouting is useful for fault tolerance
- Deflection routing (any free channel) provides probabilistic livelock freedom