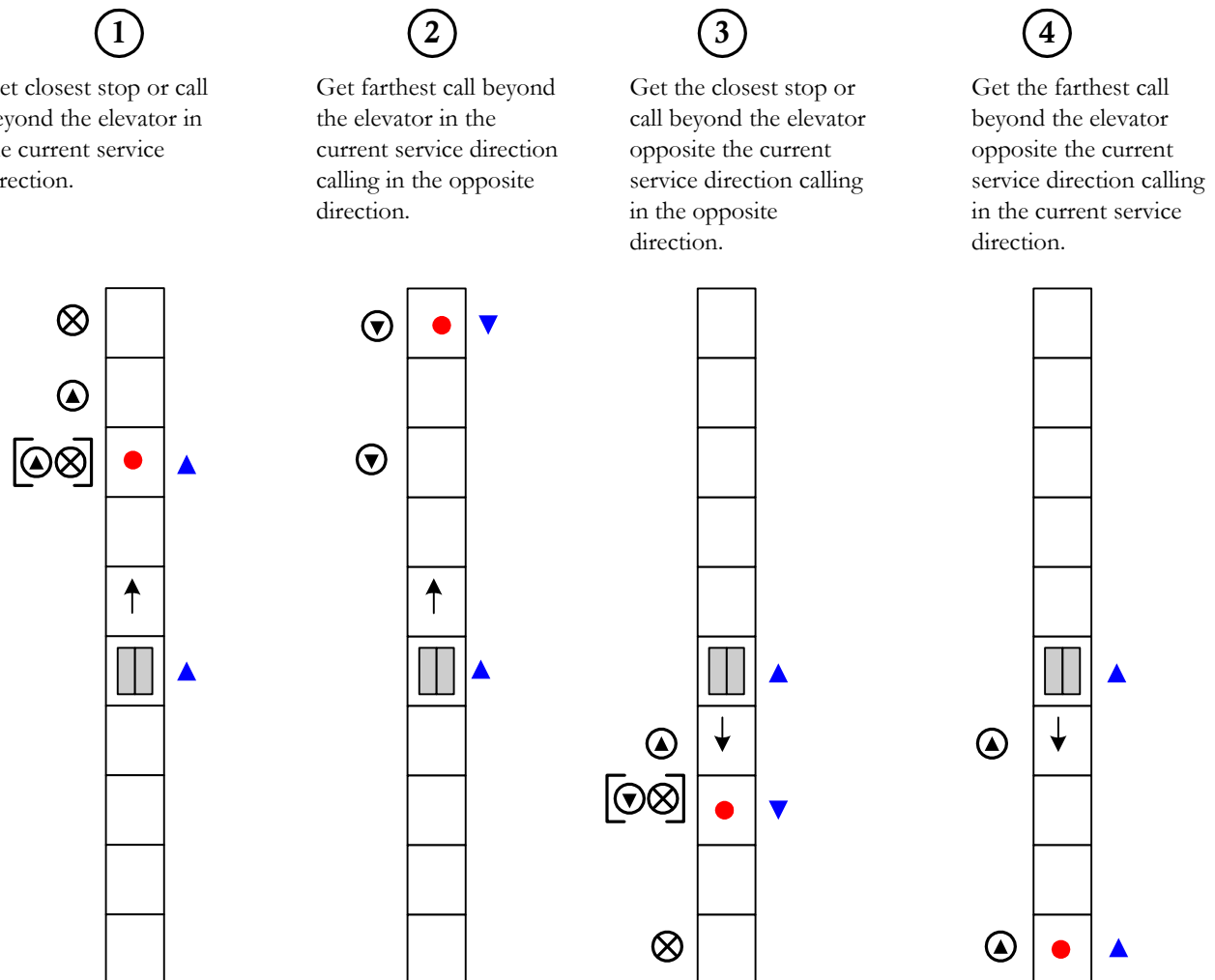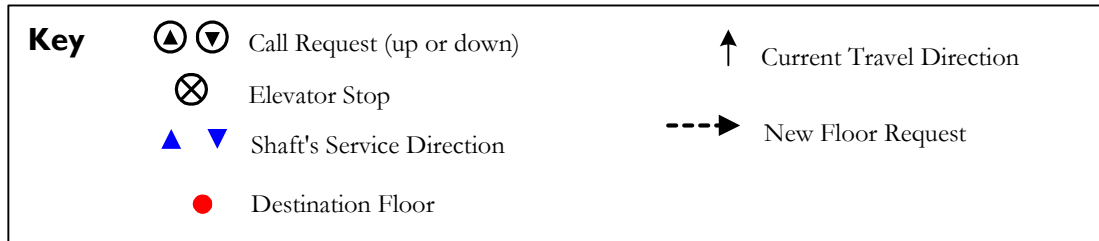# Floor Search Algorithm

The primary destination selection algorithm

Basic principle: Elevator travels as far as it usefully can in one direction, then it switches direction. If there are no requests pending, the elevator waits.

Key
- ⊙ ⊙ Call Request (up or down)
- ⊗ Elevator Stop
- ▲ ▼ Shaft's Service Direction
- ● Destination Floor
- ↑ Current Travel Direction
- ---▶ New Floor Request

① Get closest stop or call beyond the elevator in the current service direction.

② Get farthest call beyond the elevator in the current service direction calling in the opposite direction.

③ Get the closest stop or call beyond the elevator opposite the current service direction calling in the opposite direction.

④ Get the farthest call beyond the elevator opposite the current service direction calling in the current service direction.

Floor Search Algorithm

The destination may be changed while the elevator is moving

Once the elevator is on it's way - guided by a ramp managed by the Motion Control domain, new floor requests will arrive. In many cases, the new request will simply be saved and the elevator will proceed to its original destination. But sometimes it will be desirable to immediately accommodate the new request and change the current destination.

The Motion Control domain may or may not accommodate a destination change request. If the elevator is slowing down to stop at its destination, we don't want to suddenly speed up. So once the elevator speed starts ramping down, the current destination cannot be changed. Also, if the elevator is moving at high speed enroute to a distant floor and a stop is selected right in front of the direction of travel, Motion Control may decide that it cannot stop the elevator in time. So this is yet another case where we would like to change the destination, but due to Motion Control limitations, the request must be saved for later.

There are many cases where the destination should not change

Most new requests should not result in any change to the current destination. In these cases, the request is posted and the elevator continues to the current destination. The request should eventually be selected as a destination.

Destination update policy for all scenarios

We must establish a policy for each possible scenario as to whether the current elevator destination is to be updated and whether or not this affects the current service direction.
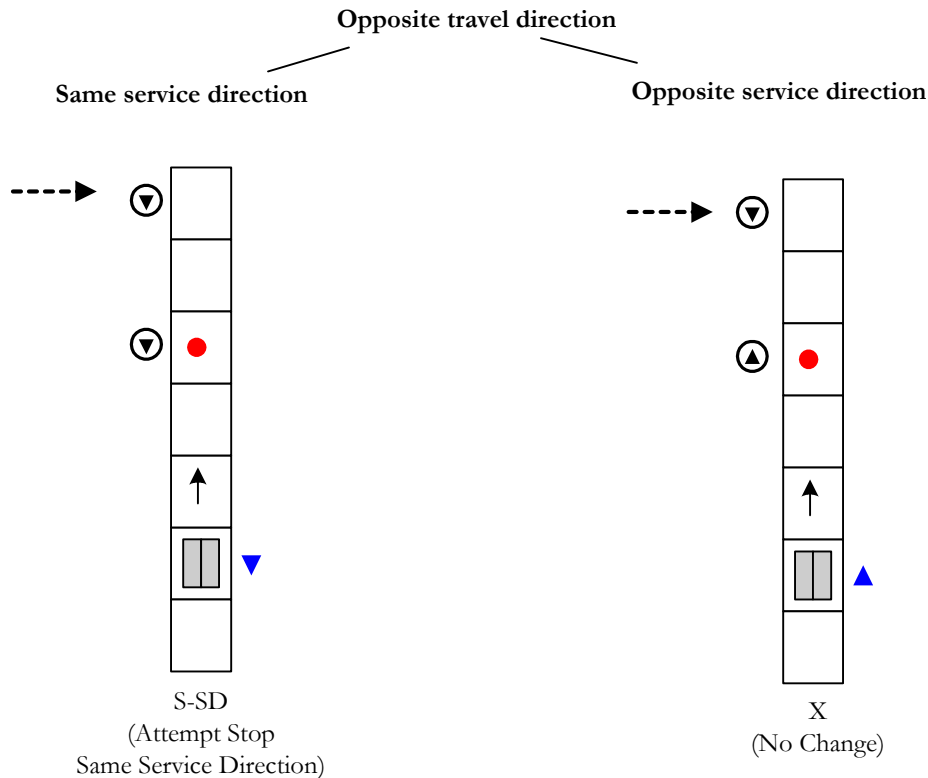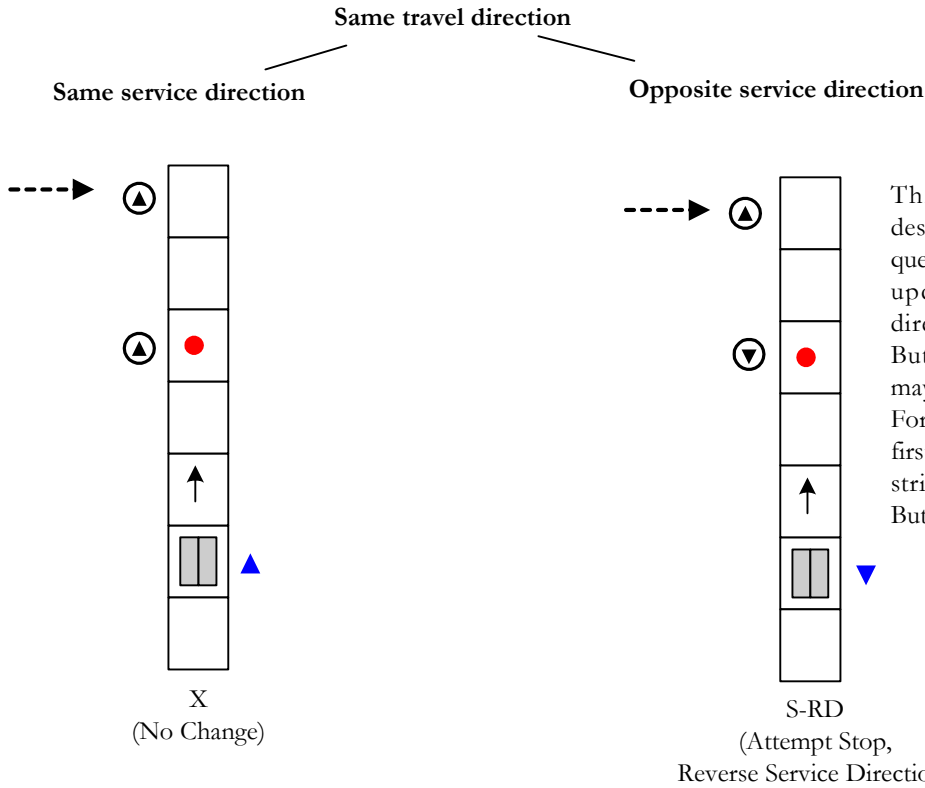
For a new request, we must consider the following characteristics:

  a) type of request (call up, call down, stop)
  b) location of the request relative to the elevator and current destination
         (between, ahead, behind)
  c) in the case of call requests the direction relative to the current service and travel direction
         (same/opposite of travel direction X same/opposite of service direction)

These scenarios are illustrated on the following pages and are then summarized in a table.
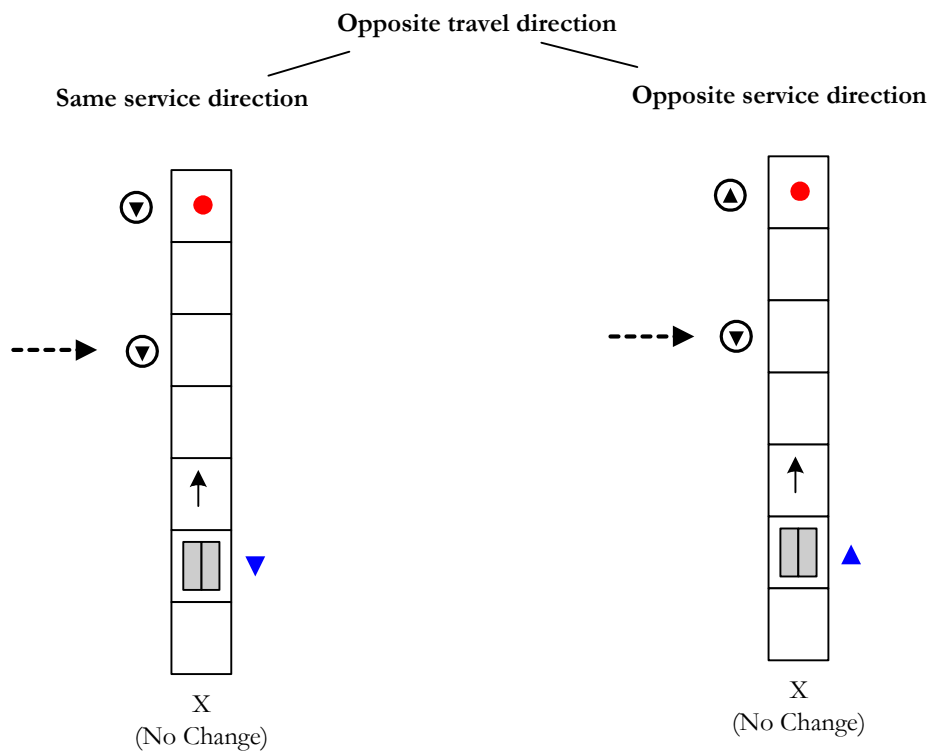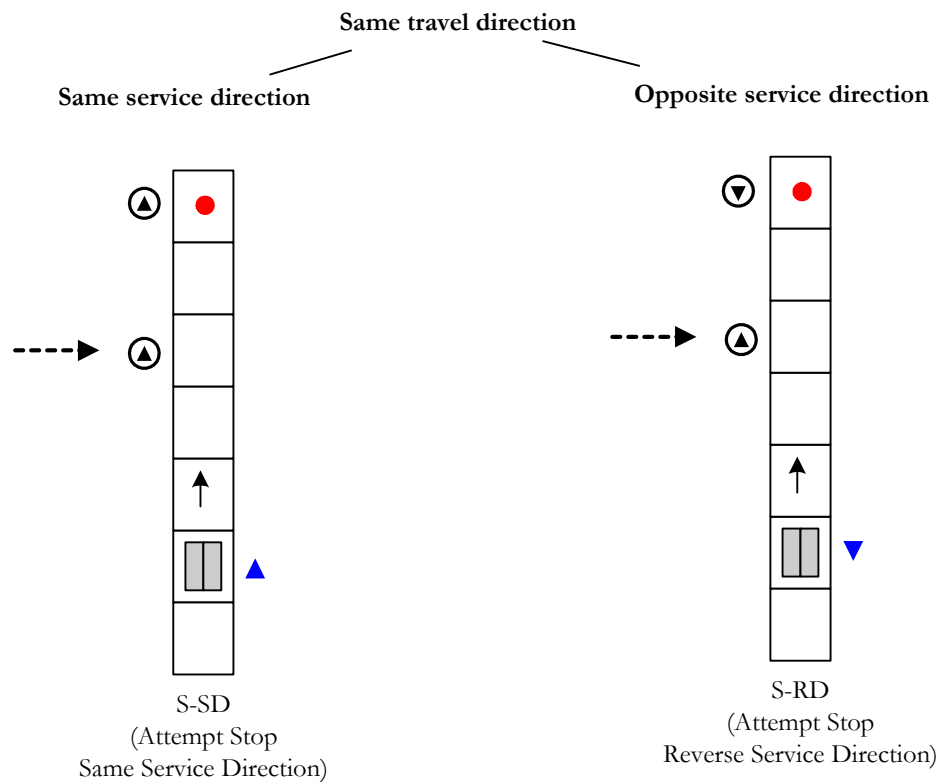
Floor Search Algorithm

Ahead of destination

Same travel direction

Same service direction

X
(No Change)

Opposite service direction

This is the only case where the primary destination selection algorithm is called into question. Strict interpretation would have us update both the destination and service direction to accommodate the new request. But if we take time waiting into account we may want to give priorty to the older request. For the sake of simplicity, and considering our first release requirements, we will adhere to the strict interpretation and update the request. But this may change in a later release.

S-RD
(Attempt Stop,
Reverse Service Direction)

Opposite travel direction

Same service direction

S-SD
(Attempt Stop
Same Service Direction)

Opposite service direction

X
(No Change)

Floor Search Algorithm

Between elevator and destination

Same travel direction

Same service direction	**Opposite service direction**

S-SD
(Attempt Stop
Same Service Direction)

S-RD
(Attempt Stop
Reverse Service Direction)

Opposite travel direction

Same service direction	**Opposite service direction**

X
(No Change)

X
(No Change)

Floor Search Algorithm

Behind elevator and destination

Same travel direction

Same service direction

X
(No Change)

Opposite service direction

X
(No Change)

Opposite travel direction

Same service direction

X
(No Change)

Opposite service direction

X
(No Change)
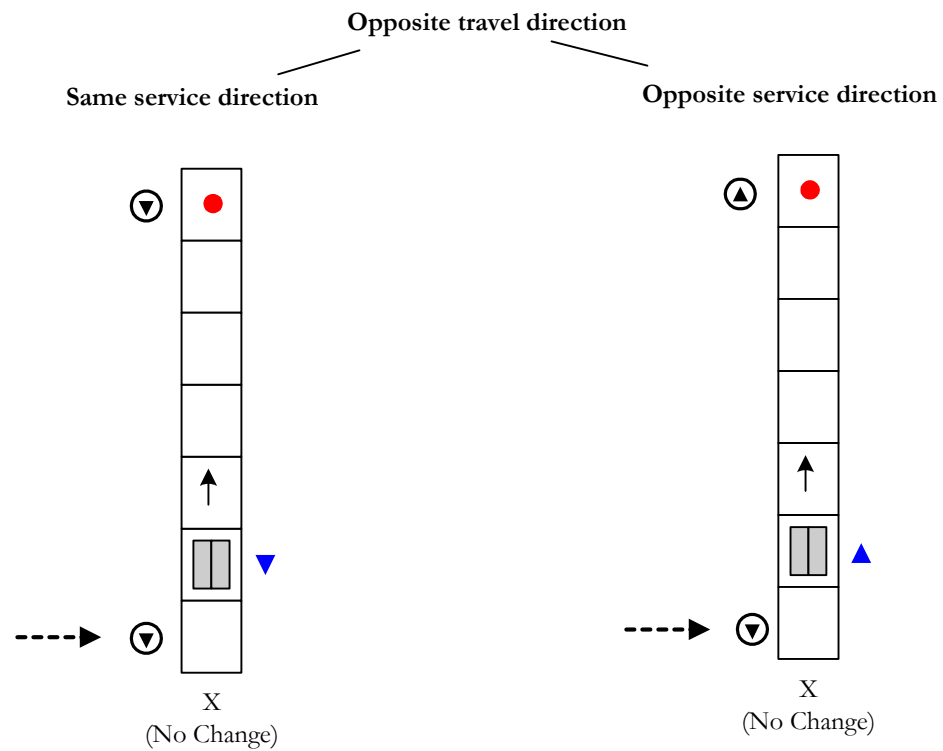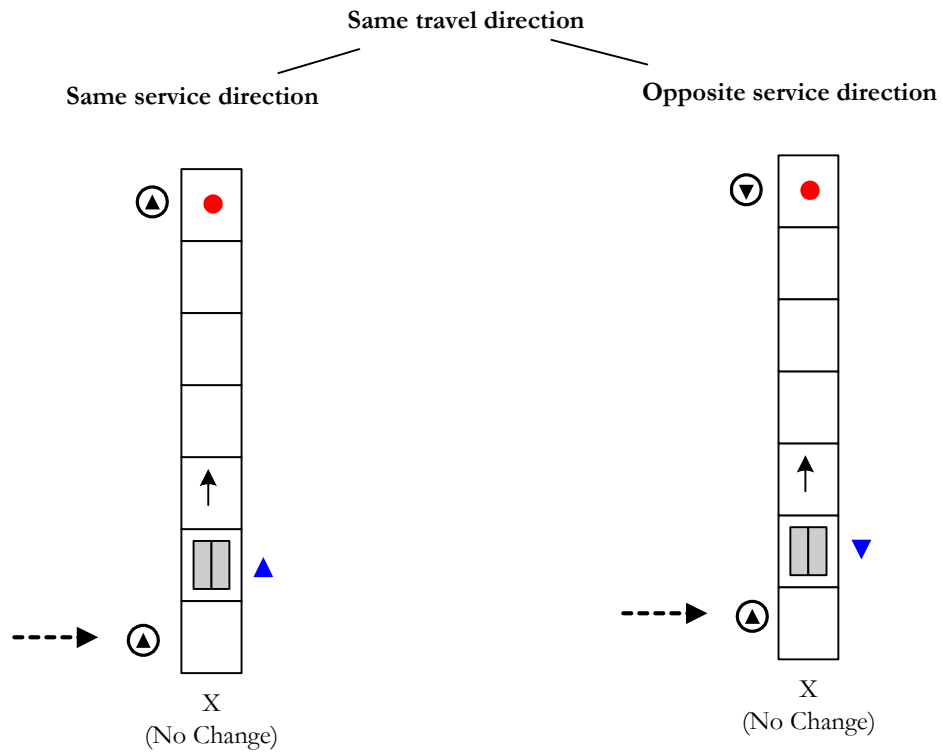
Floor Search Algorithm

Elevator Stop

Position releative to elevator and destination

Between	Ahead	Behind
X (No Change)	X (No Change)	X (No Change)
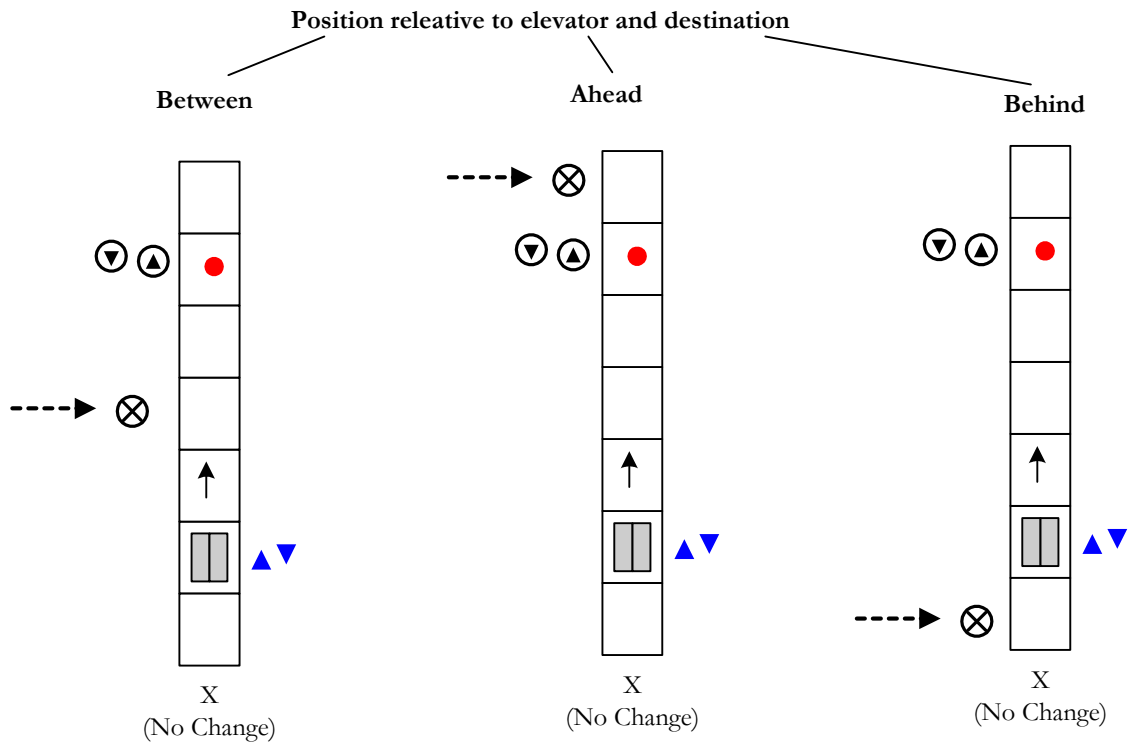
Floor Search Algorithm

The following text describes the complete set of responses to scenarios where a new request is posted while the elevator is enroute to a destination.

Call Request

Travel Direction		Same		Opposite	
Service Direction	Same	Opposite	Same	Opposite	**Stop**
Between	S-SD	S-RD	X	X	S-SD
Ahead	X	S-RD	S-SD	X	X
Behind	X	X	X	X	X

S-SD = Attempt Stop, Keep the Same Service Direction
S-RD = Attempt Stop, Reverse the Service Direction
X = Don't Change Current Destination

This table supports the scenarios shown on the previous pages.

December 21, 1999
Leon Starr
mint.elev.tn.5

Version 1.3