

下面举例说明由近及远算法的工作情况。

结点A是申请者，若只有结点F具有A要的资源，按由近及远算法，搜索过程为：

- (1) A向B发搜索信 ($p=A$)；(规则1)。
- (2) B向A发信告知，A的后结点是B；(规则2)。
- (3) B向上邻结点A发信 ($p=B$)，请继续搜索；(规则2)。
- (4) A向下邻结点C发搜索信 ($p=B$)；(规则3)。
- (5) C向B发信告知，B的后结点是C；(规则2)。
- (6) C向上邻结点A发信 ($p=C$)，请继续搜索；(规则2)。
- (7) A将继续搜索信转给它的后结点B ($p=C$)；(规则3)。
- (8) B向D发搜索信 ($p=C$)；(规则3)。
- (9) D向C发信告知，C的后结点是D；(规则2)。
- (10) D向上邻结点B发信 ($p=D$)，请继续搜索；(规则2)。
- (11) B向E发搜索信 ($p=D$)；(规则3)。
- (12) E向D发信告知，D的后结点是E；(规则3)。
- (13) E向上邻结点B发信 ($p=E$)，请继续搜索；(规则2)。
- (14) B将继续搜索信转给它的后结点C ($p=E$)；(规则3)。
- (15) C向E发搜索信 ($p=E$)；(规则3)。
- (16) E将搜索信退回给C；(规则5)。
- (17) C将继续搜索信转给它的后结点D；(规则3)。
- (18) D将继续搜索信转给它的后结点E；(规则3)。
- (19) E向F发搜索信 ($p=E$)；(规则3)。
- (20) F向E发成功信 ($p=F$)；(规则2)。
- (21) E向B转发成功信 ($p=F$)；(规则4)。
- (22) B向A转发成功信 ($p=F$)；(规则4)。
- (23) A收到来自F的成功信 ($p=F$)；(规则4)。

如果F也没有A所要的资源，则(19)步后的过程为：

- (20)' F向E告知，E的后结点是F；(规则2)。
- (21)' F向E发信请继续搜索 ($p=F$)；(规则2)。
- (22)' E将继续搜索信转给它的后结点F ($p=F$)；(规则3)。
- (23)' F既无资源又无后结点，F发失败信给上邻接点E；(规则3)。
- (24)' E向B转发失败信；(规则4)。
- (25)' B向A转发失败信；(规则4)。
- (26)' A收到失败信，搜索失败；(规则4)。

执行过程见图1。

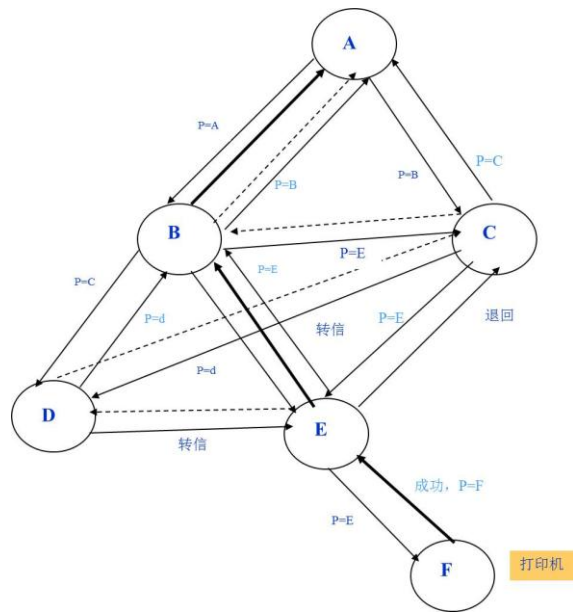


图1 由近及远算法例子

不难验证，采用由近及远算法搜索资源不会产生饥饿。被搜索到的每个结点几乎都接收到这样的三条消息，即搜索消息，通知谁是后结点的消息和由前结点转发来的继续搜索消息。因此，如果不考虑一个结点多次被搜索的情况，或者近考虑树形网络的情况，在最坏情况下需要发 $4n$ 条消息进行资源搜索工作。此外，还要加上搜索到资源后转发的成功信。因而，看起来由近及远算法比前两种算法通信量大得多。但是，当系统中有较多的结点拥有资源时，采用这种算法往往很快就能获得资源。因此，对于“稀有”资源可能招标算法或回声算法比较合适，而对于普遍拥有的资源，则由近及远算法可能更好些。

为了减少通信量，可以改进由近及远算法，但改善的算法比原来的复杂。这里不再做介绍。

上述算法略微改进就可以具有较强的强健性。我们规定，发搜索消息至一个相邻结点后，如果在 t 时间内无回答，则认为它已失效，然后向另外一个相邻结点发搜索消息或者向其后续结点转发继续搜索消息。不难验证，只要在算法执行过程中不产生新的失效结点，并且失效结点不被恢复，则增加了如上规则后，由近及远算法是强健的。