

上)。B 将忽略 (丢弃而不转发) 从任何其他结点 (如从路由器 C 或 D) 接收的源为 A 的分组。我们现在考虑结点 C, C 将直接从 A 以及从 B 接收源为 A 的分组。因为 B 不在 C 自己到 A 的最短路径上, C 将忽略来自 B 的任何源为 A 的分组。在另一方面, 当 C 接收到直接来自 A 的源为 A 的分组, 它将向结点 B、E 和 F 转发该分组。

3. 生成树广播

虽然序号控制洪泛和 RPF 避免了广播风暴, 但它们不能完全避免冗余广播分组的传输。例如, 在图 4-44 中, 结点 B、C、D、E 和 F 接收到一个或两个冗余分组。在理想情况下, 每个结点应当仅接收广播分组的一个副本。查看在图 4-45a 中由粗线相连结点组成的树, 你能看到如果广播分组仅沿着该树中的链路转发的话, 所有网络结点的每个将恰好接收到广播分组的一个副本, 这正是我们要寻找的解决方案! 该树是一棵生成树 (spanning tree) 的例子, 它包含了图中的每个结点。更形式化地讲, 图 $G=(N, E)$ 的一棵生成树是一个图 $G'=(N, E')$, 使得 E' 是 E 的子集, G' 是连通的, G' 不包含圈, 并且 G' 包括了在 G 中的所有初始结点。如果每段链路具有相应的费用且一棵树的费用就是其链路费用之和, 则在该图的所有生成树中费用最小的生成树 (毫不奇怪地) 被称为最小生成树 (minimum spanning tree)。

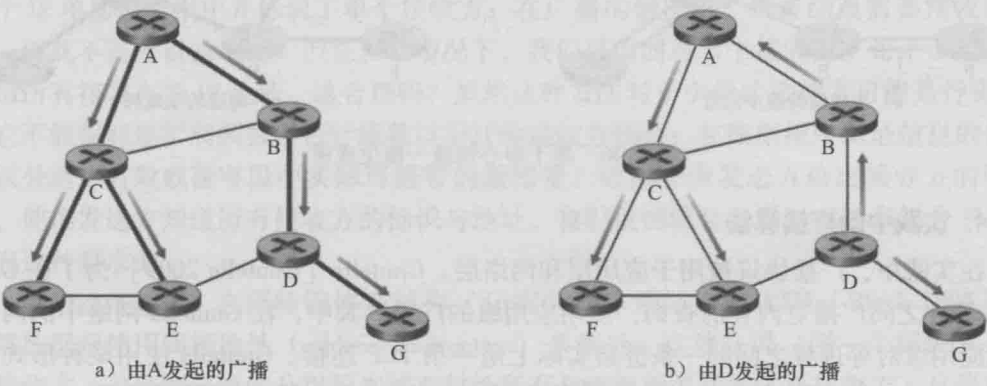


图 4-45 沿着一棵生成树的广播

因此, 提供广播的另一种方法是首先对网络结点构造出一棵生成树。当一个源结点要发送一个广播分组时, 它向所有属于该生成树的特定链路发送分组。接收广播分组的结点则向在生成树中的所有邻居转发该分组 (其接收该分组的邻居除外)。生成树不仅消除了冗余的广播分组, 而且一旦合适, 该生成树能够被任何结点用于开始广播分组, 如图 4-45a 和图 4-45b 中所示。注意到一个结点不必知道整棵树; 只需要知道它在 G 中的哪些邻居是生成树的邻居。

与生成树方法相关的复杂性主要是生成树的生成和维护。已经研制出了若干种分布式生成树算法 [Gallager 1983, Gartner 2003]。这里我们仅考虑一种简单的算法。采用基于中心的方法 (center-based approach) 建立一棵生成树时, 要定义一个中心结点 (也称之为汇合点 (rendezvous point) 或核 (core))。结点则向中心结点单播加入树 (tree-join) 报文。加入树报文使用单播路由选择朝着中心结点转发, 直到它到达一个已经属于生成树的结点或到达该中心。在任一种情况下, 加入树报文经过的路径定义了发起加入树报文的边缘结点和中心之间的生成树分支。这个新分支能够认为已被嫁接到现有的生成树上了。