我们考虑网络中有N个实时数据流

每个数据流表示为

Sk表示数据流的源节点

Dk表示数据流的目的节点

Pk，Bk表示数据流的路径（在我们这个问题中分为主路径P和备用路径B）

表示数据流的执行时间

表示数据流的截止期

表示数据流的周期

图路由的概念：



如上图所示，这是一个流从源节点a到目的节点c的图路由调度过程，a→b→c为该数据流的主路径，每段传输（即每一跳）需要2个time slot，对于主路径上的每个节点（除去目的节点）需要寻找一条备用路径到达目的节点，备用路径上的每段传输（即每一跳）只需要1个time slot. （其实这里我们可以把备用路径上的数据流作为新的数据流，它和主路径上的数据流存在冲突）

数据流冲突的概念：

对于任意两条数据流，如下图所示：



这是两个数据流

数据流1：k→h→i→l→m

数据流2：g→h→i→j→n

在无线传感网当中，每个传感节点上面都配置有一个天线，在同一时刻（一个time slot内）只能进行发送或者接收当中的一项工作（即半双工通信方式），并且也只能处理一个数据包。

对于上面这个例子，我们从数据流2的角度来看，数据流1和它有冲突的路径为

（g，h）（h，i）（i，j）.

我们的问题分为两部分：

1. 为每个数据流寻找图路由路径

对于一个给定的图.如下图所示.



在这个问题中，我们假设流都是由传感节点到网关节点的数据流，即简化了问题，源节点可以是任意的，目的节点只有网关。（上图中假设c是网关节点）。

我们有随机生成的若干流F1，F2,…FN。假设这些流已经按照RM的优先级（即周期）排好序了，优先级F1>F2>…>FN。

对于第一个流，假设其是从源节点a到目的节点c，我们为其计算源到目的的最短路径（因为是优先级最高的第一个流，所以别的流对它没有影响），作为它的主路径。假设现在数据流1的主路径是a→b→c， 在为其寻找备用路径的时候，我们需要更新网络中的链路权重，假设原来所以链路的权重都为1。因为现在数据1的主路径为a→b→c，所有与节点a，节点b，节点c有关的路径都要更新链路权重。（即图中的链路ag，ah，ab，bh，bc，bd，bf，cf，cd（这些都是无向的））。这里我们利用dijkstra算法，在为每一条数据流计算最短路径的时候，需要更新链路的权重系数，在这里，我们计算备用路径的时候，我们将上述所有路径的权重更新。

伪代码：

**Function**：FindPath(G, F)

**Input**: A graph G(V, E)， A flow set F={F1，F2，…Fn}（Fk=(sk,dk,Tk)我们假设Dk=Tk，简化问题）。

**Variable**：link weight **w**，link 延时系数 **c**

**Output：**route paths for each flow F

**Initialization： for** each link  **do**

w(u,v)=1;

c(u,v)=0;

**Begin:** **for** each flow Fk in F **do**

Find the primary path Pk connecting sk to dk；

Assign Pk as flow Fk’s primay path;

**For** each link  **do**

**If** (u,v) shares at least one node with Pk

**Then** c(u,v)=c(u,v)+2\*1/Tk

Update the weight w(u,v)=1+Tk\*c

**For** each link (u,v)Pk **do**

w(u,v)=w(u,v)+10000(这里是为了下面选择备用路径时不经过主路径)

**For** each node ik in Pk-dk **do（**为除去目的节点的其余主路径上所有节点找备用路径）

Find the back up path Bk connecting ik to dk;

Assign Bk as flow Fk’s back path;

**For** each link  **do**

**If** (u,v) shares at least one node with Bk

**Then** c(u,v)=c(u,v)+1/Tk

Update the weight w(u,v)=1+Tk\*c

**For** each link (u,v)Pk **do**

w(u,v)=w(u,v)-10000（主路径恢复正常权值）

1. 数据流按照图路由路径进行调度

**Function**：CAF-Schedule（M, F）

**Input：**Channel M，A flow set F orderd by RM with Fk=（sk，dk，Pk, Bk, Tk）

**Variable: Int** s[M][T]// a M\*T schedule table

**Output:** A schedule table

**Initialization:** Initialize all elements in s to 0; //0 for “idle”

**Begin:**

**For** each flow Fk in F **Do**

**If** Fk is a primary path flow

**For** each hop in Pk **Do**

**For** each time slot t in T **Do**

**For** each channel m in M **Do**

If (s[m][t]=0 and no shared nodes in hop) {

S[m][t]=1;

S[m][t+1]=1;

}

**End If**

**If** Fk is a backup path flow

**For** each hop in Pk **Do**

**For** each time slot t in T **Do**

**For** each channel m in M **Do**

If (s[m][t]=0 and no shared nodes in hop) {

S[m][t]=1;

}

**End If**

**Return** a schedule table S[M][T]