1.在/servlib/geohistorydtn外添加的东西，都会有//add by gaorui //end by gaorui做注释

2. . ./servlib/bundling/bundle

bundle的数据结构，根据需求，又补充了一些成员和成员函数

3. ./apps/dtnsend/dtnsend.cc

用来自定义发送bundle的app

加了一个参数 "-g" 用来输入目的地

例：

./dtnsend -s dtn://192.168.5.11.gao.com -d dtn://192.168.5.12.gao.com -t m -p “hello” -g "121412"

4. ./applib/APIServer.cc

根据densend发送的参数生成bundle

根据-g参数后的目的地id，设置bundle三层目的地id，

设置bundle为Data\_bundle，设置区域层次为0，如果-g有参数，则重新复制为3，根据区域层次判断该bundle是普通的bundle还是修改后的bundle。设置普通转发阶段bundle的副本数，以及洪泛阶段的bundle副本数，设置该bundle目前不在洪泛阶段

5. ./servlib/bundling/PrimaryBlockProcessor.cc

对于生成的bundle做了一定的处理

6../servlib/geohistorydtn/routing/GeoHistoryRouter.cc

(1)int route\_bundle(Bundle\* bundle);

转发bundle,

如果bundle类型是NEI\_AREA\_BUNDLE，则交给(3)处理

如果是原DTN2bundle或修改后的bundle但目的结点为本结点，

则不做其他处理，按原本DTN2处理

如果是修改后，目的地不是本结点，

a.判断当前邻居是否是bundle的目的结点，如果是，直接交付

b.判断当前结点是否在目的区域，如果是，执行洪泛，即将每一个邻居结点都转发一半副本，如果副本个数小于1，则不转发

c.不是，执行普通转发

挑选出离目的地最近的link，将该link离目的地最近的区域和本节点离目的地最近的区域放入到一变量中，进行机会值排序(ChanceValueSort::getAllAvaliableNodeArea)，排序后根据顺序进行转发，如果排名高的就转发一半，低就转发1个

(2)void handle\_routeAllBundle();

目前没有，王伟涛遗留

(3)int route\_neighbourArea\_bundle(Bundle \*bundle);

转发类型为NEI\_AREA\_BUNDLE的bundle

(4) bool canDirectDelivery(Bundle \*bundle);

如果当前的邻居有一个是该bundle的目的结点，则直接交付

(5)int get\_matching\_RouteEntryVec

(Bundle \*bundle,RouteEntryVec &entry\_vec,

int sameAreaLevel,

map<Arearef,RouteEntryref> &sameLevelAreaMap);

根据参数，挑选合适的link

如果是洪泛，将选中所有的link

如果是普通，则选择离目的地最近的邻居link，仅仅一个

如果bundle是NEI\_AREA\_BUNDLE,则挑选特定的link

7.机会值计算

ChanceValueSort/ChanceValueCompute

(1)ChanceValueSort::getAllAvaliableNodeArea

总函数，将历史邻居Area加入到已有当前邻居和本节点离目的地最近的Area的list形参中，计算机会值，并排序，后返回排序后的各Area

(2) ChanceValueSort::addNeibourAreaNode

将List形参中已有的当前邻居和本节点离目的地最近的Area计算机会值

遍历历史邻居，将每一个邻居离离目的地最近的Area加入到list中，并计算历史邻居的机会值

(3) ChanceValueCompute::carryChance(bundle，Area)

计算机会值

得到bundle有效的生命周期，然后计算Area所属的每一层的机会值，返回

(4) ChanceValueCompute::carryChance(validvector，Area)

计算机会值

将生命周期与Area中的频率向量相乘，相乘后再相加，得到每一个时间尺度下的机会值sum，最后将每一个时间尺度下的机会值相加，返回

(5)历史邻居的机会值与当前的邻居的机会值的差别在于

将Neighbour的其中一个尺度的频率向量与生命周期相乘，再相加，最后除以尺度(60,24等)，得到的这个数与sum相乘即可