数据库设计规划

### 数据库应该完成的功能

1. 存储抓包结果

存储抓包得到的原始数据，包括编号、SSRC编号、时间戳timestamp、毫秒级时间与数据包序号对应、丢失数据包序号、创建时间、更新时间

1. 存储调制源数据

数据消息，包括编号、待发送数据长度、数据的二进制表现形式、创建时间、更新时间

参数配置及对应的数据表，包括编号、参数组合

1. 存储调制解调结果

调制解调结果集合，包括编号、抓包编号、消息编号、参数配置列表、调制中丢弃数据包、解调数据的二进制表示、传输数据位数、错误数据位数、创建时间、更新时间

1. 存储分布统计结果

包括原始的分布情况及调制后的分布情况，包括编号、相对BL、IPD分布结果、连续丢包数量的分布、丢包时间间隔的分布

包括分布结果相似性评价的量化评估值，包括编号、相对BL、K-S测试结果、KLD测试结果、T-test结果等

1. 存储视频质量评估结果

包括原始的相对视频质量损失及调制后的相对视频质量损失，包括编号、相对BL、PSNR值、SSIM值、FSIM值

### 数据库分库及分表设计

1. `source` 数据库

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| captured | | | | | | | | | |
| ID | TYPE | INFO | MAXSEQ | SSRC | TIMESTAMP | TIMEMAP | LOSTLIST | UPDATETIME | CREATETIME |
| INT | INT | TEXT | INT | INT | INT | LONGTEXT | LONGTEXT | TIMESTAMP | TIMESTAMP |
| CREATE TABLE `source`.`captured` (  `ID` INT NOT NULL AUTO\_INCREMENT COMMENT '数据编号' ,  `TYPE` INT NULL DEFAULT 1 COMMENT '噪声类型，1表示实际的，2表示生成的，3表示分场景的' ,  `INFO` TEXT CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NULL DEFAULT NULL COMMENT '数据信息备注',  `MAXSEQ` INT UNSIGNED NULL DEFAULT NULL COMMENT '最大的数据包序号',  `SSRC` INT UNSIGNED NULL DEFAULT NULL COMMENT 'SSRC值的十进制表示' ,  `TIMESTAMP` INT UNSIGNED NULL DEFAULT NULL COMMENT 'TIMESTAP的十进制表示' ,  `TIMEMAP` LONGTEXT NULL DEFAULT NULL COMMENT '数据包序号与毫秒级时间的映射，JSON表示' ,  `LOSTLIST` LONGTEXT NULL DEFAULT NULL COMMENT '丢失数据包序号的列表，JSON列表表示' ,  `UPDATETIME` TIMESTAMP on update CURRENT\_TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '数据更新时间' ,  `CREATETIME` TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '数据创建时间' ,  PRIMARY KEY (`ID`),  INDEX `id` (`ID`),  INDEX `type` (`TYPE`)  ) ENGINE = InnoDB  CHARSET=utf8mb4  COLLATE utf8mb4\_unicode\_ci  COMMENT = '抓包数据列表'; | | | | | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| data | | | | |
| ID | LENGTH | DATA | UPDATETIME | CREATETIME |
| INT | INT | LONGTEXT | TIMESTAMP | TIMESTAMP |
| CREATE TABLE `source`.`data` (  `ID` INT NOT NULL AUTO\_INCREMENT COMMENT '数据索引' ,  `LENGTH` INT UNSIGNED NULL DEFAULT NULL COMMENT '数据最大长度' ,  `DATA` LONGTEXT CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NULL DEFAULT NULL COMMENT '数据的01形式表示' ,  `UPDATETIME` TIMESTAMP on update CURRENT\_TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '数据条目更新时间' ,  `CREATETIME` TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '数据条目创建时间' ,  PRIMARY KEY (`ID`),  INDEX `id` (`ID`)  ) ENGINE = InnoDB  CHARSET=utf8mb4  COLLATE utf8mb4\_unicode\_ci  COMMENT = '发送数据列表'; | | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| parameters | | | | | | | |
| ID | LCODEWORD | LHASH | LCRC | R | MCOLS | UPDATETIME | CREATETIME |
| INT | INT | INT | INT | INT | INT | TIMESTAMP | TIMESTAMP |
| CREATE TABLE `source`.`parameters` (  `ID` INT NOT NULL AUTO\_INCREMENT COMMENT '数据索引' ,  `LCODEWORD` INT UNSIGNED NULL DEFAULT NULL COMMENT '码字长度参数' ,  `LHASH` INT UNSIGNED NULL DEFAULT NULL COMMENT 'HASH校验长度参数' ,  `LCRC` INT UNSIGNED NULL DEFAULT NULL COMMENT 'CRC校验长度参数' ,  `R` INT UNSIGNED NULL DEFAULT NULL COMMENT '校验复位参数' ,  `MCOLS` INT UNSIGNED NULL DEFAULT NULL COMMENT '编码矩阵的列数' ,  `UPDATETIME` TIMESTAMP on update CURRENT\_TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '数据更新时间' ,  `CREATETIME` TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '数据创建时间' ,  PRIMARY KEY (`ID`),  INDEX `id` (`ID`)  ) ENGINE = InnoDB  CHARSET=utf8mb4  COLLATE utf8mb4\_unicode\_ci  COMMENT = '参数数据表'; | | | | | | | |

1. `modulation`数据库

数据表的命名，按照PARAID的方式进行命名，其中，参数PARAID为参数配置组合的序号。

假定待发送数据条目为10，原始测试数据数量为30。对于方案一，参数BL的取值范围为6-14，R的取值范围为1-10 ，则每个表记录数量为10，表数量为30\*9\*10=2700；对于方案二，参数Lcodeword的取值范围，为6-14，参数Lhash的取值范围为Lcodeword-2，参数Lcrc的取值范围为Lcodeword-Lhash-1，R的取值范围为1-10，Mcols的取值范围为2、4、6、8、10、……40共20个，则每个参数下的最大数量，为10条，数据表数量为9\*9\*9\*10\*20\*30=4374000。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| PARAID | | | | | | | | | |
| ID | PARAID | SOURCEID | DATAID | DROPLIST | DEMODULATED | SENTBITS | ERRORBITS | UPDATETIME | CREATETIME |
| INT | INT | INT | INT | LONGTEXT | LONGTEXT | INT | INT | TIMESTAMP | TIMESTAMP |
| CREATE TABLE `modulation`.`1\_1` (  `ID` INT NOT NULL AUTO\_INCREMENT COMMENT '数据索引' ,  `PARAID` INT NOT NULL COMMENT '参数配置的ID' ,  `SOURCEID` INT NOT NULL COMMENT '噪声数据的ID序号' ,  `DATAID` INT NOT NULL COMMENT '发送数据的ID序号' ,  `DROPLIST` LONGTEXT CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NULL DEFAULT NULL COMMENT '调制后的丢包序列' ,  `DEMODULATED` LONGTEXT CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NULL DEFAULT NULL COMMENT '解调得到的二进制序列' ,  `SENTBITS` INT UNSIGNED NOT NULL DEFAULT '0' COMMENT '发送位数' ,  `ERRORBITS` INT UNSIGNED NOT NULL DEFAULT '0' COMMENT '错误位数' ,  `UPDATETIME` TIMESTAMP on update CURRENT\_TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '数据更新时间' ,  `CREATETIME` TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '数据创建时间' ,  PRIMARY KEY (`ID`),  INDEX `data` (`DATAID`),  INDEX `id` (`ID`),  INDEX `source` (`SOURCEID`),  INDEX `para` (`PARAID`)  ) ENGINE = InnoDB  CHARSET=utf8mb4  COLLATE utf8mb4\_unicode\_ci  COMMENT = '调制解调详情表'; | | | | | | | | | |

1. `distribution`数据库

分布统计数据表命名，SOURCEID\_distribution，包含的字段：ID、PARAID、IPDDIS、COUNTDIS、GAPDIS、UPDATETIME、CREATETIME。量化评估数据表命名， SOURCEID\_evaluation，包含的字段：ID、DISID、KSP、KLD、TTESTP、 UPDATETIME、CREATETIME。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| distribution\_SOURCEID | | | | | | | | | |
| ID | SOURCEID | PARAID | IPDDIS | COUNTDIS | WIN100DIS | …… | WIN500DIS | UPDATETIME | CREATETIME |
| INT | INT | INT | LONGTEXT | LONGTEXT | LONGTEXT | …… | LONGTEXT | TIMESTAMP | TIMESTAMP |
| CREATE TABLE `distribution`.`1\_distribution` (  `ID` INT NOT NULL AUTO\_INCREMENT COMMENT '数据索引' ,  `SOURCEID` INT NOT NULL COMMENT '抓包序列的ID' ,  `PARAID` INT NOT NULL COMMENT '参数配置的ID' ,  `IPDDIS` LONGTEXT CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NULL DEFAULT NULL COMMENT 'IPD分布' ,  `COUNTDIS` LONGTEXT CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NULL DEFAULT NULL COMMENT '连续丢包数分布' ,  `WIN100DIS` LONGTEXT CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NULL DEFAULT NULL COMMENT '100数据包窗口丢包数量的统计分布' ,  ……  `WIN1000DIS` LONGTEXT CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NULL DEFAULT NULL COMMENT '1000数据包窗口丢包数量的统计分布' ,  `UPDATETIME` TIMESTAMP on update CURRENT\_TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '数据更新时间' ,  `CREATETIME` TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '数据创建时间' ,  PRIMARY KEY (`ID`),  INDEX `para` (`PARAID`),  INDEX `id` (`ID`),  INDEX `source` (`SOURCEID`)  ) ENGINE = InnoDB  CHARSET=utf8mb4  COLLATE utf8mb4\_unicode\_ci  COMMENT = '分布信息统计表'; | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| cdf\_SOURCEID | | | | | | | | | |
| ID | SOURCEID | PARAID | IPDCDF | COUNTCDF | WIN100CDF | …… | WIN1000CDF | UPDATETIME | CREATETIME |
| INT | INT | INT | LONGTEXT | LONGTEXT | LONGTEXT | …… | LONGTEXT | TIMESTAMP | TIMESTAMP |
| CREATE TABLE `distribution`.`cdf\_1` (  `ID` INT NOT NULL AUTO\_INCREMENT COMMENT '数据索引' ,  `SOURCEID` INT NOT NULL COMMENT '抓包序列的ID' ,  `PARAID` INT NOT NULL COMMENT '参数配置的ID' ,  `IPDCDF` LONGTEXT CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NULL DEFAULT NULL COMMENT 'IPD分布的累积分布函数' ,  `COUNTCDF` LONGTEXT CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NULL DEFAULT NULL COMMENT '连续丢包数分布的累积分布函数' ,  `WIN100CDF` LONGTEXT CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NULL DEFAULT NULL COMMENT '100数据包窗口丢包数量的累积分布函数' ,  ……  `WIN1000CDF` LONGTEXT CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NULL DEFAULT NULL COMMENT '1000数据包窗口丢包数量的累积分布函数' ,  `UPDATETIME` TIMESTAMP on update CURRENT\_TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '数据更新时间' ,  `CREATETIME` TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '数据创建时间' ,  PRIMARY KEY (`ID`),  INDEX `para` (`PARAID`),  INDEX `id` (`ID`),  INDEX `source` (`SOURCEID`)  ) ENGINE = InnoDB  CHARSET=utf8mb4  COLLATE utf8mb4\_unicode\_ci  COMMENT = '分布信息的累积分布函数表'; | | | | | | | | | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| pmf\_SOURCEID | | | | | | | | | |
| ID | SOURCEID | PARAID | IPDPMF | COUNTPMF | WIN100PMF | …… | WIN1000PMF | UPDATETIME | CREATETIME |
| INT | INT | INT | LONGTEXT | LONGTEXT | LONGTEXT | …… | LONGTEXT | TIMESTAMP | TIMESTAMP |
| CREATE TABLE `distribution`.`pmf\_1` (  `ID` INT NOT NULL AUTO\_INCREMENT COMMENT '数据索引' ,  `SOURCEID` INT NOT NULL COMMENT '抓包序列的ID' ,  `PARAID` INT NOT NULL COMMENT '参数配置的ID' ,  `IPDPMF` LONGTEXT CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NULL DEFAULT NULL COMMENT 'IPD分布的概率密度函数' ,  `COUNTPMF` LONGTEXT CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NULL DEFAULT NULL COMMENT '连续丢包数分布的概率密度函数' ,  `WIN100PMF` LONGTEXT CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NULL DEFAULT NULL COMMENT '100数据包窗口丢包数量的概率密度函数' ,  ……  `WIN1000PMF` LONGTEXT CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci NULL DEFAULT NULL COMMENT '1000数据包窗口丢包数量的概率密度函数' ,  `UPDATETIME` TIMESTAMP on update CURRENT\_TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '数据更新时间' ,  `CREATETIME` TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '数据创建时间' ,  PRIMARY KEY (`ID`),  INDEX `para` (`PARAID`),  INDEX `id` (`ID`),  INDEX `source` (`SOURCEID`)  ) ENGINE = InnoDB  CHARSET=utf8mb4  COLLATE utf8mb4\_unicode\_ci  COMMENT = '分布信息的概率密度函数表'; | | | | | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| evaluation\_SOURCEID | | | | | |
| ID | SOURCEID | PARAID | KSP-IPD, KLD-IPD, TTESTP-IPD, ES-IPD, WASSERSTEIN-IPD, ENERGY-IPD  KSP-WIN100, KLD-WIN100, TTESTP-WIN100, ES-WIN100, WASSERSTEIN-WIN100, ENERGY- WIN100  KSP-WIN200, KLD-WIN200, TTESTP-WIN200, ES-WIN200, WASSERSTEIN-WIN200, ENERGY- WIN200  KSP-WIN500, KLD-WIN500, TTESTP-WIN500, ES-WIN500, WASSERSTEIN-WIN500, ENERGY- WIN500 | UPDATETIME | CREATETIME |
| INT | INT | INT | TIMESTAMP | TIMESTAMP |
| CREATE TABLE `distribution`.`evaluation \_1` (  `ID` INT NOT NULL AUTO\_INCREMENT COMMENT '数据索引' ,  `SOURCEID` INT NOT NULL COMMENT '抓包序列的ID' ,  `PARAID` INT NOT NULL COMMENT '参数配置的ID' ,  `KSP-IPD` DOUBLE NOT NULL COMMENT 'IPD分布K-S测试的p值' ,  `KLD-IPD` DOUBLE NOT NULL COMMENT 'IPD分布KLD测试的熵值' ,  `TTESTP-IPD` DOUBLE NOT NULL COMMENT 'IPD分布Ttest的p值' ,  `ES-IPD` DOUBLE NOT NULL COMMENT 'IPD分布ES测试的p值' ,  `WASSERSTEIN-IPD` DOUBLE NOT NULL COMMENT 'IPD分布WASSERSTEIN 距离' ,  `ENERGY-IPD` DOUBLE NOT NULL COMMENT 'IPD分布ENERGY距离' ,  `KSP-WIN100` DOUBLE NOT NULL COMMENT '100窗口丢包数分布K-S测试的p值' ,  `KLD-WIN100` DOUBLE NOT NULL COMMENT '100窗口丢包数分布KLD测试的熵值' ,  `TTESTP-WIN100` DOUBLE NOT NULL COMMENT '100窗口丢包数分布Ttest的p值' ,  `ES-WIN100` DOUBLE NOT NULL COMMENT '100窗口丢包数分布ES测试的p值' ,  `WASSERSTEIN-WIN100` DOUBLE NOT NULL COMMENT '100窗口丢包数分布WASSERSTEIN 距离' ,  `ENERGY-WIN100` DOUBLE NOT NULL COMMENT '100窗口丢包数分布ENERGY距离' ,  `KSP-WIN200` DOUBLE NOT NULL COMMENT '200窗口丢包数分布K-S测试的p值' ,  `KLD-WIN200` DOUBLE NOT NULL COMMENT '200窗口丢包数分布KLD测试的熵值' ,  `TTESTP-WIN200` DOUBLE NOT NULL COMMENT '200窗口丢包数分布Ttest的p值' ,  `ES-WIN200` DOUBLE NOT NULL COMMENT '200窗口丢包数分布ES测试的p值' ,  `WASSERSTEIN-WIN200` DOUBLE NOT NULL COMMENT '200窗口丢包数分布WASSERSTEIN 距离' ,  `ENERGY-WIN200` DOUBLE NOT NULL COMMENT '200窗口丢包数分布ENERGY距离' ,  `KSP-WIN500` DOUBLE NOT NULL COMMENT '500窗口丢包数分布K-S测试的p值' ,  `KLD-WIN500` DOUBLE NOT NULL COMMENT '500窗口丢包数分布KLD测试的熵值' ,  `TTESTP-WIN500` DOUBLE NOT NULL COMMENT '500窗口丢包数分布Ttest的p值' ,  `ES-WIN500` DOUBLE NOT NULL COMMENT '500窗口丢包数分布ES测试的p值' ,  `WASSERSTEIN-WIN500` DOUBLE NOT NULL COMMENT '500窗口丢包数分布WASSERSTEIN 距离' ,  `ENERGY-WIN500` DOUBLE NOT NULL COMMENT '500窗口丢包数分布ENERGY距离' ,  `UPDATETIME` TIMESTAMP on update CURRENT\_TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '记录的更新时间' ,  `CREATETIME` TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '记录的创建时间' ,  PRIMARY KEY (`ID`),  INDEX `id` (`ID`)  ) ENGINE = InnoDB  CHARSET=utf8mb4  COLLATE utf8mb4\_unicode\_ci  COMMENT = '分布统计量化评估结果表'; | | | | | |

几种检测方式的详细说明：

1. K-S测试：Kolmogorov-Smirnov测试，p值高于0.05则表示分布一致；scipy.stats.ks\_2samp测试的对象，是两个分布的原始分布数据；
2. KLD测试：Kullback-Leibler divergence，entropy熵值低于0.1，则分布一致；scipy.stats.entropy双样本测试，对象为两个分布的概率密度PMF；
3. TTEST测试：Welch’s t-test测试，p值高于0.05则表示分布一致；scipy.stats.ttest\_ind指定equal\_var为false触发，测试对象，为两个分布的原始数据；
4. ES测试：Epps-Singleton测试，对比于K-S测试，K-S测试假设用于连续分布数据，ES测试面向离散分布场景；scipy.stats.epps\_singleton\_2samp的测试对象，是两个样本的观测值，也就是原始分布数据；
5. WASSERSTEIN距离：Wasserstein距离，测试结果表示两个分布之间的差距，scipy.stats.wasserstein\_distance测试对象为两个分布的概率密度函数PMF；
6. ENERGY距离：Energy距离，测试两个分布之间的差异，scipy.stats.energy\_distance的测试对象，是概率密度函数PMF。
7. `evaluation`数据库

视频质量评估的数据表，命名方式SOURCEID\_videoquality，包含的字段：ID、PARAID、PSNR、SSIM、FSIM、UPDATETIME、CREATETIME。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| SOURCEID\_videoquality | | | | | | |
| ID | PARAID | PSNR | SSIM | FSIM | UPDATETIME | CREATETIME |
| INT | INT | DOUBLE | DOUBLE | DOUBLE | TIMESTAMP | TIMESTAMP |
| CREATE TABLE `evaluation`.`1\_videoquality` (  `ID` INT NOT NULL AUTO\_INCREMENT COMMENT '数据索引' ,  `SOURCEID` INT NOT NULL COMMENT '源数据的ID' ,  `PARAID` INT NOT NULL COMMENT '参数配置的ID' ,  `PSNR` DOUBLE NULL DEFAULT NULL COMMENT '平均PSNR值' ,  `SSIM` DOUBLE NULL DEFAULT NULL COMMENT '平均SSIM值' ,  `FSIM` DOUBLE NULL DEFAULT NULL COMMENT '平均FSIM值' ,  `UPDATETIME` TIMESTAMP on update CURRENT\_TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '记录更新时间' ,  `CREATETIME` TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP COMMENT '记录创建时间' ,  PRIMARY KEY (`ID`),  INDEX `para` (`PARAID`),  INDEX `id` (`ID`)  ) ENGINE = InnoDB  CHARSET=utf8mb4  COLLATE utf8mb4\_unicode\_ci  COMMENT = '视频质量评价结果表'; | | | | | | |

### 数据库并行及互斥

**统计分布特征**

每个表，面向一个噪声序列，则数据表中的每个条目，表示的的不同调制参数下，调制结果的分布情况。

首先，获取特定类型噪声的所有ID，作为创建表的依据；生成全局变量，作为线程检索过程的参考顺序；

接下来，获取调制数据库中所有数据表，调制数据库中数据表的命名，是按照参数进行的，也就是说，同一个表中的差异只有隐蔽消息及噪声ID；

在ProcessOperation的Process过程，创建distribution中的数据表，数据表的命名，按照噪声的序号进行，同时清空数据表中的数据，并为原始分布创建统计结果，默认的PARAID为0，调用的函数，与Task执行线程一致。生成全局变量，映射SOURCEID及丢包List。生成全局变量，映射SOURCEID及数据包的时间戳。

准备工作完成，开始创建执行线程，Task执行线程的创建划分，是modulation数据库的数据表命名；

Task执行线程，按照噪声ID依次进行，首先筛选出SOURCEID对应的所有DROPLIST；按照DROPLIST条目，依次进行统计任务，并将统计结果汇总到一个MAP映射中；也就是，每个modulation数据表中，针对特定的噪声ID，只对应distribution表中的一条记录；

统计过程，分三种类型，一种是IPD，一种是连续丢包的数量，一种是连续无丢包的数量，分别以三种函数进行实现；函数需要的参数，包括SOURCEID，QLIST \*DropList，QMap &result；

最终完成，将结果插入到数据库。

**一致性评价指标**

添加每100/200/400/500/1000区间的平均丢包率、丢包个数统计

连续丢包数、丢包间隔，更改为100，200，500区间内的丢包数量统计

需要测试的分布统计，应当满足在不同的场景中，基本都能通过，也就是分场景特征，要符合整体特征。改变一下思路，统计规律针对的是整体，那么，可以部分替换，用调制后的部分规律，替换调制前的部分规律，然后再与原始整体规律进行对比。