Precoded BATS codes

Shenghao Yang
Institute of Network Coding
The Chinese University of Hong Kong, Hong Kong SAR, China
Email: shyang@inc.cuhk.edu.hk

August 24, 2012

Contents

Abstract

Chapter 1

Algorithms

1.1 To Do

1. The number of LDPC symbols should be prime.
2. De-randomness of the degree and index generation. Refer to [**?**, §5.3.5].
3. De-randomness of the generator matrices. Refer to §**??**.
4. NC coder starts after receiving enough number of packets of a batch.

1.2 Precoded Packets

1.2.1 Parity-Check Matrix

The precode is systematic and can be described using a parity-check matrix $\mathbf{P}$, which is an $(K + L + H) \times (L + H)$ matrix. (For a sequence x of the precode, we have $x\mathbf{P} = 0$.)

The first L columns of $\mathbf{P}$ is for LDPC. L is a prime number that is approximately $K\eta + \sqrt{2K}$. The BATS code is designed to recover at least $(1 - \eta)K$ packets.

The last H columns of $\mathbf{P}$ is for HDPC. H is between 10 and 16, and H grows very slowly as a function of K.

We now describe the submatrices of the parity-check matrix in details. Let $\mathbf{B}_A$ be the active input packets, $\mathbf{B}_I$ be the inactive input packets, $\mathbf{L}$ be the LDPC packets and $\mathbf{H}$ be the HDPC packets. The parity check constraint is given by

$$\begin{bmatrix}\mathbf{B}_A & \mathbf{L} & \mathbf{B}_I & \mathbf{H}\end{bmatrix} \begin{bmatrix}\mathbf{C}_A & \mathbf{Q}_A \\ \mathbf{I}_L & \mathbf{Q}_L \\ \mathbf{C}_I & \mathbf{Q}_I \\ \mathbf{C}_H & \mathbf{I}_H\end{bmatrix} = \mathbf{0}.$$

The first L columns have four parts. Here $\mathbf{C}_A$ containts circulant matrices. For $i = 0, 1, \ldots,$ the first row of the ith circulant matrix has ones at positions 0, $(i+1)$ mod (L), and $(2i+2)$ mod (L) and zeros elsewhere. The other columns are cyclic shifts (right-shift) of the first.

$\mathbf{I}_L$ is an identity matrix.

Let $\mathbf{C}_P = \begin{bmatrix} \mathbf{C}_I \\ \mathbf{C}_H \end{bmatrix}$. Columns of $\mathbf{C}_P$ are cyclic and each column consists of two consecutive ones.

The last H columns consist of two parts. The last H rows are a $H \times H$ identity matrix. Let

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_A \\ \mathbf{Q}_L \\ \mathbf{Q}_I \end{bmatrix}.$$

The matrix $\mathbf{Q}$ is constructed such that there is an efficient algorithm for the multiplication $x\mathbf{Q}$ for a generic vector x. More explicitly, the matrix $\mathbf{Q}$ is given as $\mathbf{Q} = \Gamma\Delta$, where

$$\Gamma = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{H-1} \\ 0 & 1 & \alpha & \cdots & \alpha^{H-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix},$$

and Δ is a $K + L \times H$ matrix with all rows has degree 2 (with ones in two random postions) except that the last row is given by

$$\begin{bmatrix} 1 & \alpha & \alpha^2 \cdots \alpha^{H-1} \end{bmatrix}.$$

1.2.2 Precoded Packets Generation

That is

$$\begin{bmatrix} \mathbf{L} & \mathbf{H} \end{bmatrix} \begin{bmatrix} \mathbf{I}_L & \mathbf{Q}_L \\ \mathbf{C}_H & \mathbf{I}_H \end{bmatrix} = \begin{bmatrix} \mathbf{B}_A & \mathbf{B}_I \end{bmatrix} \begin{bmatrix} \mathbf{C}_A & \mathbf{Q}_A \\ \mathbf{C}_I & \mathbf{Q}_I \end{bmatrix}$$

or

$$\begin{bmatrix} \mathbf{L} & \mathbf{H} \end{bmatrix} \begin{bmatrix} \mathbf{I}_L & \mathbf{0} \\ \mathbf{C}_H & \mathbf{I}_H + \mathbf{C}_H\mathbf{Q}_L \end{bmatrix} = \begin{bmatrix} \mathbf{B}_A & \mathbf{B}_I \end{bmatrix} \begin{bmatrix} \mathbf{C}_A & \mathbf{Q}_A + \mathbf{C}_A\mathbf{Q}_L \\ \mathbf{C}_I & \mathbf{Q}_I + \mathbf{C}_I\mathbf{Q}_L \end{bmatrix}$$

To compute $\mathbf{L}$ and $\mathbf{H}$, we can first compute $\mathbf{B}^* = \mathbf{B}_A\mathbf{C}_A + \mathbf{B}_I\mathbf{C}_I$. Then $\mathbf{H}$ is obtained by solving

$$\mathbf{H}(\mathbf{I}_H + \mathbf{C}_H\mathbf{Q}_L) = \begin{bmatrix} \mathbf{B}_A & \mathbf{B}^* & \mathbf{B}_I \end{bmatrix} \begin{bmatrix} \mathbf{Q}_A \\ \mathbf{Q}_L \\ \mathbf{Q}_I \end{bmatrix}$$

Last, $\mathbf{L} = \mathbf{H}\mathbf{C}_H + \mathbf{B}^*$.

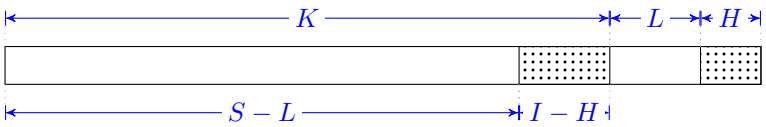

Figure 1.1: Packets partition

1.3 BATS Code

After precoding, we have K data packets, L LDPC packets and H HDPC packets. Precoded packets are partitioned into SM packets and IN packets. The number of SM packets is S and the number of IN packets is I.

1.3.1 Generator Matrices of Batches

The generator matrix of a batch with degree d is a $d \times M$ matrix.

Though random generator matrices work well, we can still try to use deterministic generator matrices, which has the advantage in term of less randomness. For degree $d < q$, we can try to use a submatrix of Vandermonde matrix as a generator matrix.

Let

$$V = \begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{M-1} \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^{M-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_d & \alpha_1^2 & \cdots & \alpha_d^{M-1} \end{bmatrix}$$

where α_i, $i = 1, \ldots d$, are different non-zero field elements of $GF(q)$.

1.4 Inactivation Decoding

1.4.1 Received Packets Processing

The received packets of a batch i are given by

$$\begin{aligned} \mathbf{Y}[i] &= \mathbf{B}[i]\mathbf{G}[i]\mathbf{H}[i] \\ &= \mathbf{B}_D[i]\mathbf{G}_D[i]\mathbf{H}[i] + \mathbf{B}_A[i]\mathbf{G}_A[i]\mathbf{H}[i] + \mathbf{B}_I[i]\mathbf{G}_I[i]\mathbf{H}[i] \end{aligned}$$

where $\mathbf{B}_D$, $\mathbf{B}_A$ and $\mathbf{B}_I$ are decoded, active but undecoded and inactive packets. When a new packet $\mathbf{Y}[i;k]$ with coding vector $\mathbf{H}[i;k]$ is received,

1. compute the active coefficients $\mathbf{C}_A[i;k] \leftarrow \mathbf{G}_A[i]\mathbf{H}[i;k]$,
2. compute the inactive coefficients $\mathbf{C}_I[i;k] \leftarrow \mathbf{G}_I[i]\mathbf{H}[i;k]$, and
3. compute the decoded coefficients $\mathbf{G}_D[i]\mathbf{H}[i;k]$ and substitute the decoded packets $\mathbf{B}_D[i] = \mathbf{Y}_{\mathbf{B}_D[i]} + \mathbf{B}_I\mathbf{C}_{\mathbf{B}_D[i]}$ as $\mathbf{Y}[i;k] \leftarrow \mathbf{Y}[i;k] - \mathbf{Y}_{\mathbf{B}_D[i]}\mathbf{G}_D[i]\mathbf{H}[i;k]$ and $\mathbf{C}_I[i;k] \leftarrow \mathbf{C}_I[i;k] + \mathbf{C}_{\mathbf{B}_D[i]}\mathbf{G}_D[i]\mathbf{H}[i;k]$.

After the above computation and substitution, the effective received packets are

$$\mathbf{Y}[i] = \mathbf{B}_A[i]\mathbf{C}_A[i] + \mathbf{B}_I\mathbf{C}_I[i]$$

where $\mathbf{B}_I$ is all the inactive packets. Note that $\mathbf{B}_A[i]$ can be empty.

1.4.2 Decoding a Batch

If $\mathbf{C}_A[i]$ has full row rank, we can decode the batch by applying Gaussian elimination such that $\mathbf{C}_A[i] \rightarrow \begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}$ and $\mathbf{Y}[i] = \mathbf{B}_A[i]\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix} + \mathbf{B}_I\mathbf{C}_I[i]$. Let

$$\begin{aligned}\mathbf{Y}_1[i] &= \mathbf{B}_A[i] + \mathbf{B}_I\mathbf{C}_1[i] \\ \mathbf{Y}_2[i] &= \mathbf{B}_I\mathbf{C}_2[i].\end{aligned}$$

After decoding all the batches, we obtain

$$\begin{bmatrix}\mathbf{B}_A & \mathbf{B}_I\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0} & \mathbf{Q}_A \\ \mathbf{C}_1 & \mathbf{C}_2 & \mathbf{Q}_I\end{bmatrix} = \begin{bmatrix}\mathbf{Y}_1 & \mathbf{Y}_2 & \mathbf{0}\end{bmatrix}$$

where the $\mathbf{Q}_A$ and $\mathbf{Q}_I$ are submatrices of $\mathbf{Q}$, which corresponds to the HDPC constraints.

1.4.3 Decoding of Inactive Packets

We can solve $\mathbf{B}_I$ by the following equation

$$\mathbf{B}_I \underbrace{\begin{bmatrix}\mathbf{C}_2 & \mathbf{C}_1\mathbf{Q}_A - \mathbf{Q}_I\end{bmatrix}}_{\mathbf{C}} = \underbrace{\begin{bmatrix}\mathbf{Y}_2 & \mathbf{Y}_1\mathbf{Q}_A\end{bmatrix}}_{\mathbf{Y}}.$$

Only linearly independant columns of $\mathbf{C}_2$ are useful.

We can compute $\mathbf{C}_1\mathbf{Q}_A + \mathbf{Q}_I$ and $\mathbf{Y}_1\mathbf{Q}_A$ using

$$\begin{aligned}\mathbf{C}_1\mathbf{Q}_A + \mathbf{Q}_I &= \begin{bmatrix}\mathbf{C}_1 & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{Q}_A \\ \mathbf{Q}_I\end{bmatrix} \\ \mathbf{Y}_1\mathbf{Q}_A &= \tilde{\mathbf{Y}}_1\mathbf{Q},\end{aligned}$$

where if the ith variable node is invative, the ith column of $\tilde{\mathbf{C}}_1$ is all zero, the ith column of $\mathbf{1}_I$ contains one 1 in the position of the inactSeq of the ith variable node, and the ith column of $\tilde{\mathbf{Y}}_1$ is all zero; if the ith variable node is decoded, the ith column of $\tilde{\mathbf{C}}_1$ is the corresponding column in $\mathbf{C}_1$, the ith column of $\mathbf{1}_I$ is all zero, and the ith column of $\tilde{\mathbf{Y}}_1$ is the corresponding column in $\mathbf{Y}_1$.

An arbitrary matrix multiplying with $\mathbf{Q}$ can be computed easily.

1.5 Other

1.5.1 Inductive Checking of Linear Independance

Let x_i, $i = 1, 2, \cdots$, be a sequence of vecotrs and B_i, $i = 0, 1, \cdots$, be a set of vectors defined as

$$\begin{aligned} B_0 &= \emptyset \\ B_i &= B_{i-1} \text{ if } x_i \not\perp B_{i-1} \\ B_i &= B_{i-1} \cup \{x_i\} \text{ if } x_i \perp B_{i-1} \end{aligned}$$

How to fast check whether $x_i \perp B_{i-1}$ or not?

This algorithm is used in NCCoder and Decoder when receiving packets, as wll as in inactive decoding.

Chapter 2

Implementations

2.1 BatsBasic Class

2.1.1 Member Variables

2.1.1.1 smNum

2.1.1.2 packets

```
SymbolType * packets;
```

Not allocated.

2.1.1.3 checkPackets

```
SymbolType * checkPackets;
```

Not allocated.

2.2 BatsEncoder Class

2.2.1 Member Functions

2.2.1.1 setInputPackets

Flow chart in Fig. **??**.

2.2.1.2 genLdpc

2.2.1.3 genHdpc

2.2.1.4 genBatchWithKey

Flow chart in Fig. **??**.

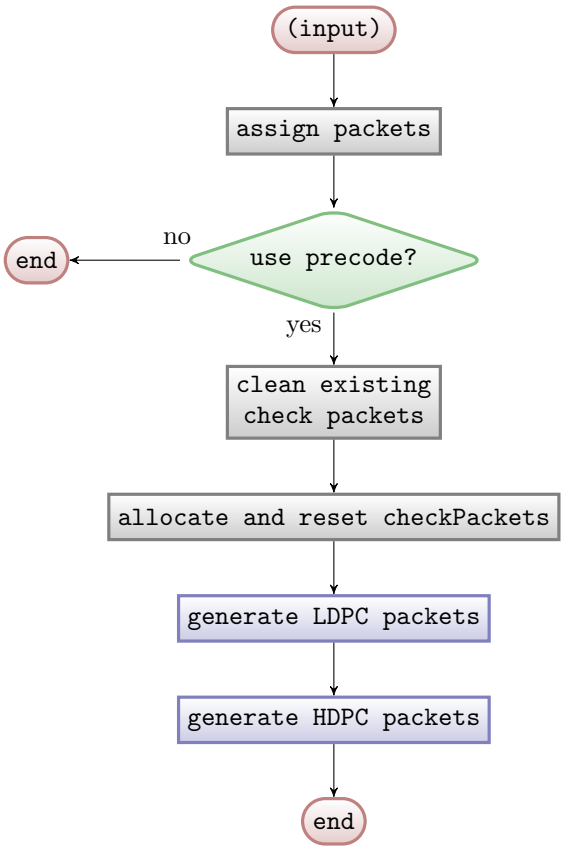

Figure 2.1: BatsEncoder::setInputPackets

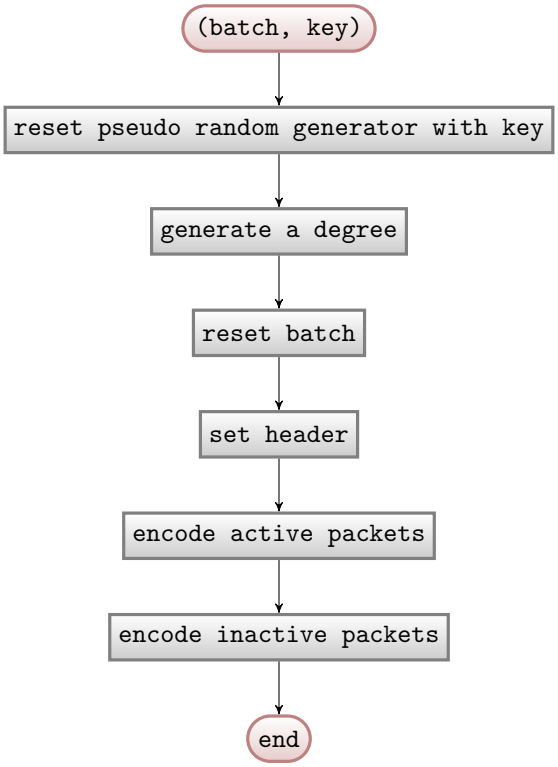

Figure 2.2: BatsEncoder::genBatchWithKey

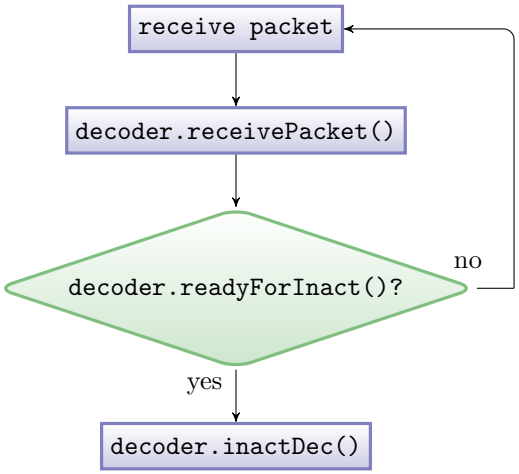

Figure 2.3: A typical flow to use an instance decoder of BatsDecoder.

2.3 VariableNode Class

2.3.1 Member Variables

2.3.1.1 *inactSeq*

The initial value of *inactSeq* is -1. For IN packets, their *inactSeq* are assigned to be from 0 to $piNum - 1$. When a new variable node is inactivated, its *inactSeq* is the current number of inactive variable nodes (excluding itself).

In the matrix $\mathbf{B}_I$, the ith column corresponds to the variable node with $inactSeq = i$. So *inactSeq* is used to access the corresponding inactivation coefficients.

2.4 CheckNode Class

2.5 BatsDecoder Class

2.5.1 Using a Decoder

Fig. **??** illustrates a type flow to perform inactivation decoding.

2.5.2 Member Variables

2.5.2.1 nRecPkg

Number of the packets received. Used to decide when to start inactivation decoding.

2.5.2.2 nRecBatch

An upper bound on the number of the batches received. Used to get the rank distribution of received batches.

2.5.2.3 nC2, maxC2, C and Y

Ref Section **??**. C and Y are matrices of maxInact+maxRedun columns. The first maxC2 (= maxInact+maxRedun - hdpcNum) columns of C and Y are used for $\mathbf{C}_2$ and $\mathbf{Y}_2$, respectively. nC2 $\leq$ maxC2 is the actual number of columns in C_2. The i column of $\mathbf{C}_2$ is saved in C[maxC2-1-i].

2.5.2.4 Decoder Status

The following are some constant of decoder.

$totalNum$ the total number of packets including input and check packets.

The following are variables of decoder status.

$nRecPkg$ the number of packets received (after checking linear independance).

$nDecoded$ the number of decoded packets (including check packets).

$nC2$ the number of packets in C_2 (after checking linear independance).

$nInactVar$ the number of inactive nodes.

2.5.3 Member Functions

2.5.3.1 receivePacket

Flow chart is in Fig. **??**.

2.5.3.2 initNewBatch

This function is similar to BatsEncoder::genBatchWithKey.

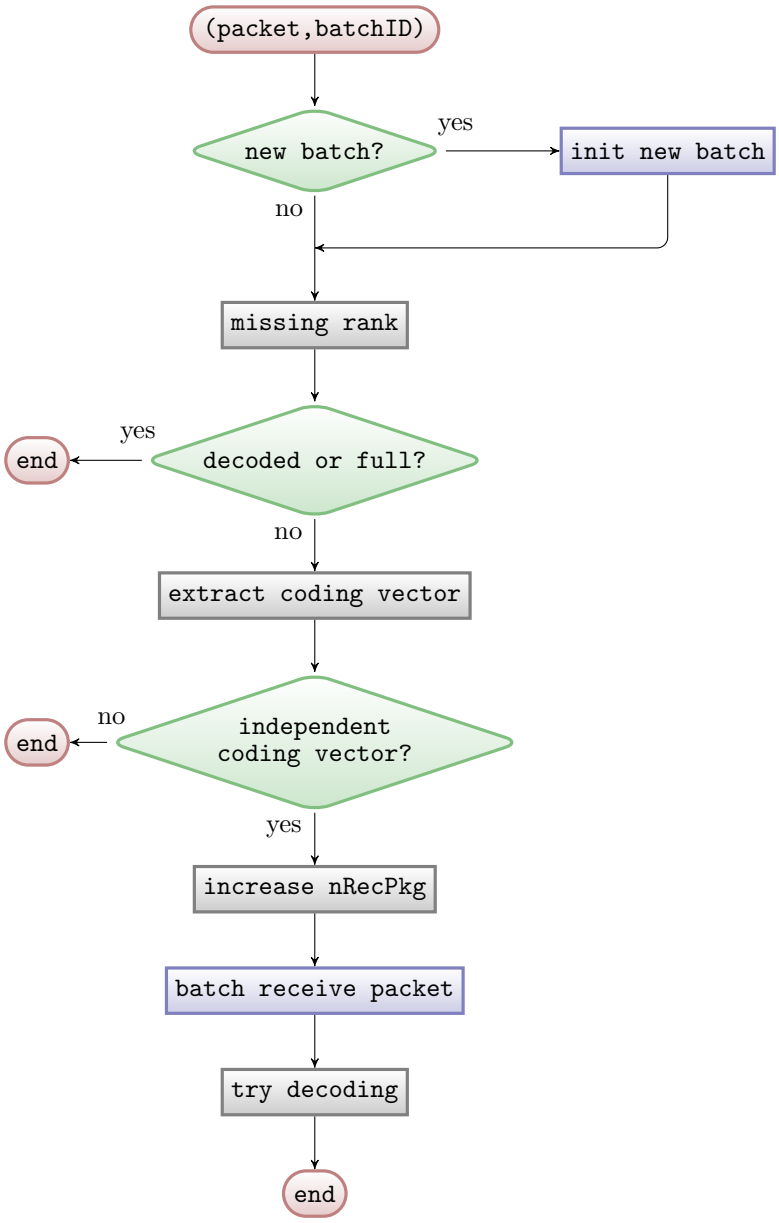

Figure 2.4: BatsDecoder::receivePacket

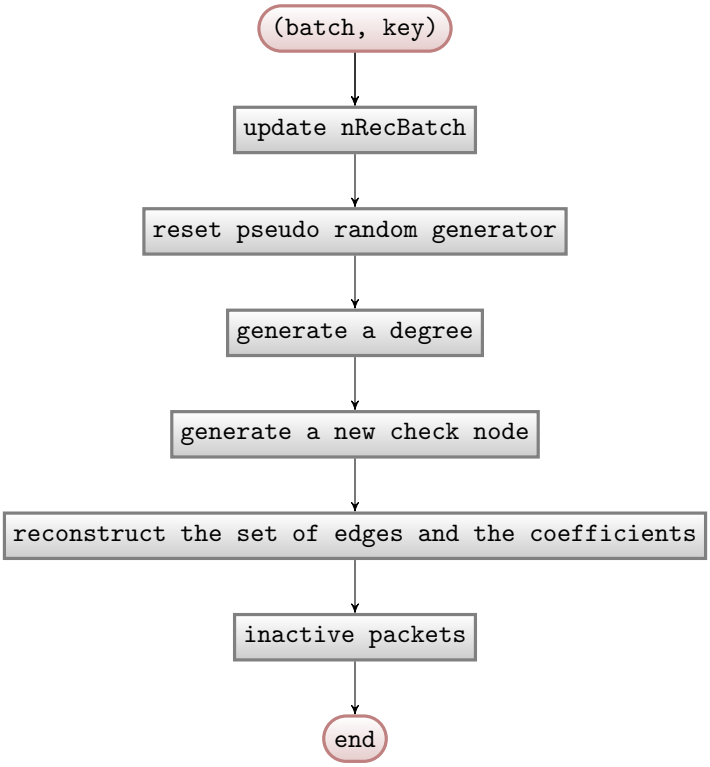

Figure 2.5: BatsDecoder::initNewBatch

Chapter 3

Testing

3.1 No HDPC and PI

In class BatsBasic, set $hdpcNum = 0$ and $piNum = hdpcNum$.

3.2 To Fix

3.2.1 Reduce Packet Drop

Received packets of decoded batches can still be put in C_2.

3.2.2 LDPC

Now there are many undecoded LDPC symbols after all batches are decoded. We can try use smaller degree of LDPC symbols, or using the idea of batched LDPC.

3.2.3 Redesign CheckNode

Bibliography

[1] http://tools.ietf.org/pdf/rfc6330.pdf