File Synchronization from Insertions and Deletions

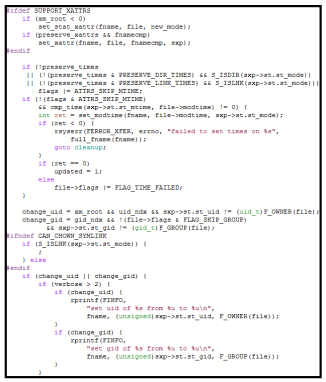
Nicolas Bitouzé
http://bitouze.bol.ucla.edu
LORIS Lab, UCLA

Joint work with:
Fred Sala (UCLA),
Lara Dolecek (UCLA),
S. M. Sadegh Tabatabaei Yazdi (Qualcomm)

CoDESS Kickoff Meeting
Sept. 19th, 2013

Motivation

Original File



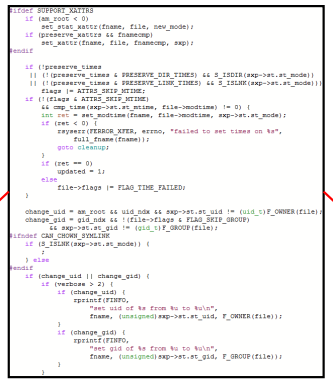
```
#ifdef SUPPORT_XATTRS
    if (am_root < 0)
        set_stat_xattr(fname, file, new_mode);
    if (preserve_xattrs && fnamecmp)
        set_xattr(fname, file, fnamecmp, sxp);
#endif

    if (!preserve_times
     || (!(preserve_times & PRESERVE_DIR_TIMES) && S_ISDIR(sxp->st.st_mode))
     || (!(preserve_times & PRESERVE_LINK_TIMES) && S_ISLNK(sxp->st.st_mode)))
        flags |= ATTRS_SKIP_MTIME;
    if (!(flags & ATTRS_SKIP_MTIME)
     && cmp_time(sxp->st.st_mtime, file->modtime) != 0) {
        int ret = set_modtime(fname, file->modtime, sxp->st.st_mode);
        if (ret < 0) {
            rsyserr(FERROR_XFER, errno, "failed to set times on %s",
                    full_fname(fname));
            goto cleanup;
        }
        if (ret == 0)
            updated = 1;
        else
            file->flags |= FLAG_TIME_FAILED;
    }

    change_uid = am_root && uid_ndx && sxp->st.st_uid != (uid_t)F_OWNER(file);
    change_gid = gid_ndx && !(file->flags & FLAG_SKIP_GROUP)
        && sxp->st.st_gid != (gid_t)F_GROUP(file);
#ifndef CAN_CHOWN_SYMLINK
    if (S_ISLNK(sxp->st.st_mode)) {
        ;
    } else
#endif
    if (change_uid || change_gid) {
        if (verbose > 2) {
            if (change_uid) {
                rprintf(FINFO,
                        "set uid of %s from %u to %u\n",
                        fname, (unsigned)sxp->st.st_uid, F_OWNER(file));
            }
            if (change_gid) {
                rprintf(FINFO,
                        "set gid of %s from %u to %u\n",
                        fname, (unsigned)sxp->st.st_gid, F_GROUP(file));
            }
        }
```

Motivation

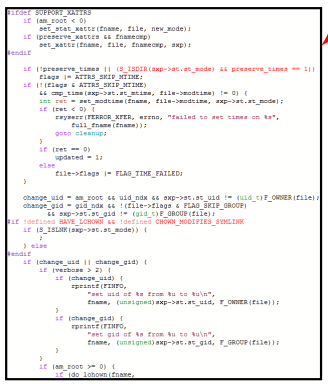
Original File



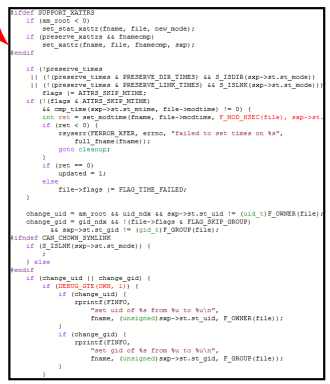
Edits

Edits

Alice's Version

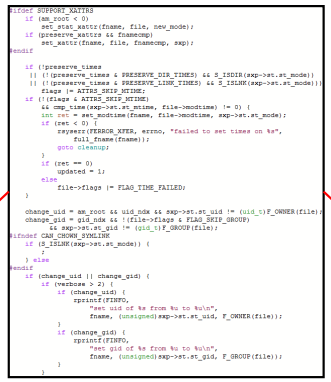


Bob's Version



Motivation

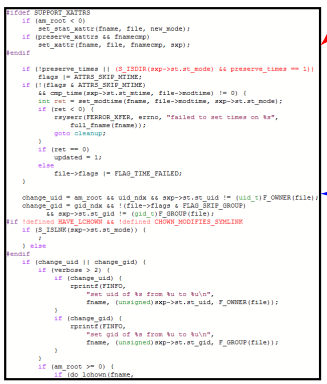
Original File



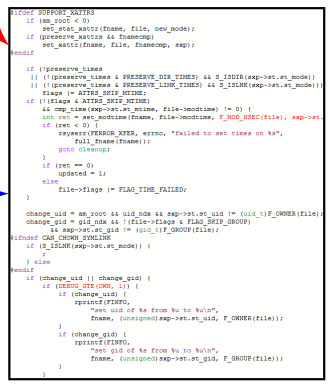
Edits

Edits

Alice's Version



Bob 's Version

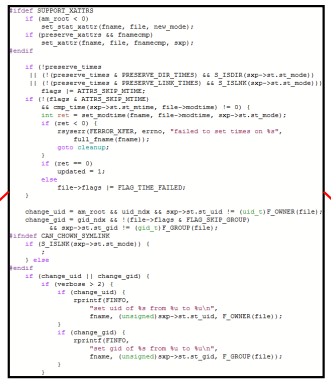


Synchronization Protocol

so that Bob's version matches Alice's

Motivation

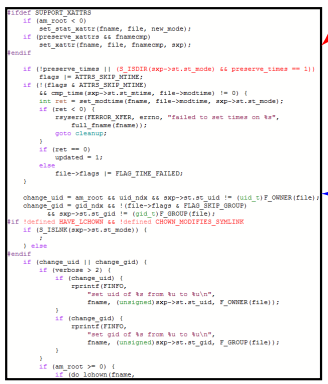
Original File



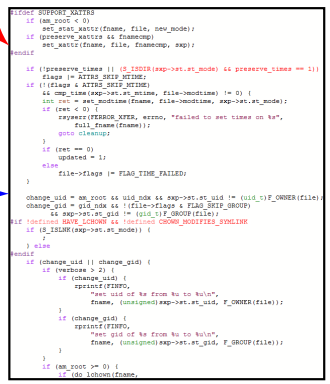
Edits

Edits

Alice's Version



Alice's Version



Synchronization Protocol

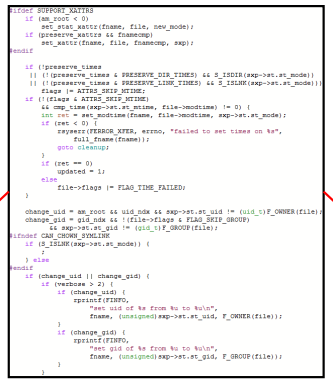
so that Bob's version matches Alice's

Motivation

Applications to:

- Cloud Synchronization (e.g., Dropbox),

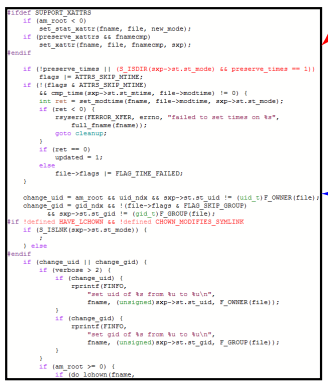
Original File



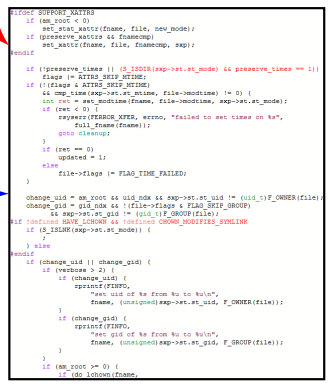
Edits

Edits

Alice's Version



Alice's Version



Synchronization Protocol

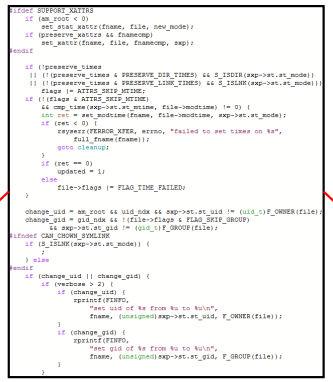
so that Bob's version matches Alice's

Motivation

Applications to:

- Cloud Synchronization (e.g., Dropbox),
- Version Control (e.g., Github, SVN),

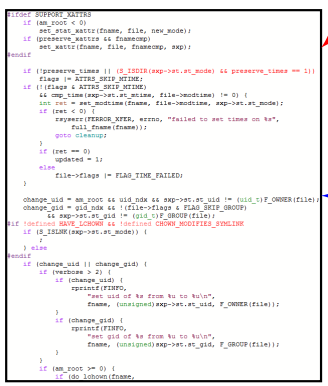
Original File



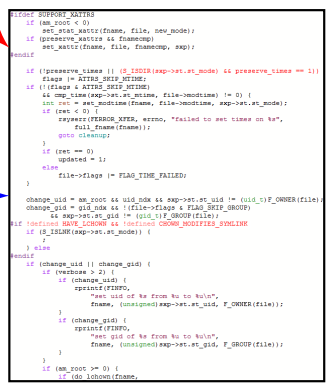
Edits

Edits

Alice's Version



Alice's Version



Synchronization Protocol

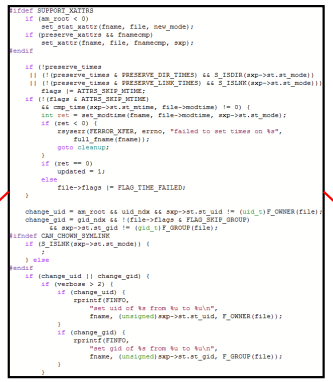
so that Bob's version matches Alice's

Motivation

Applications to:

- Cloud Synchronization (e.g., Dropbox),
- Version Control (e.g., Github, SVN),
- etc.

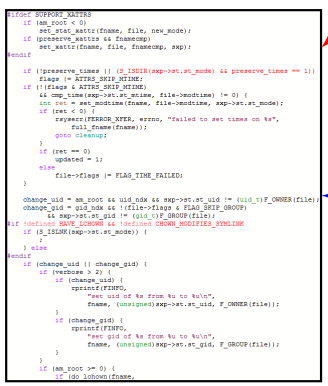
Original File



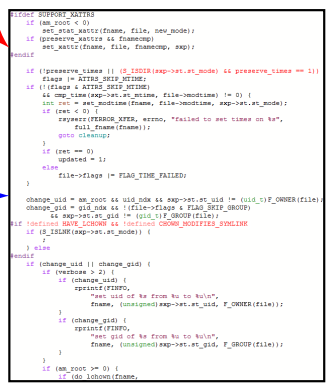
Edits

Edits

Alice's Version



Alice's Version



Synchronization Protocol

so that Bob's version matches Alice's

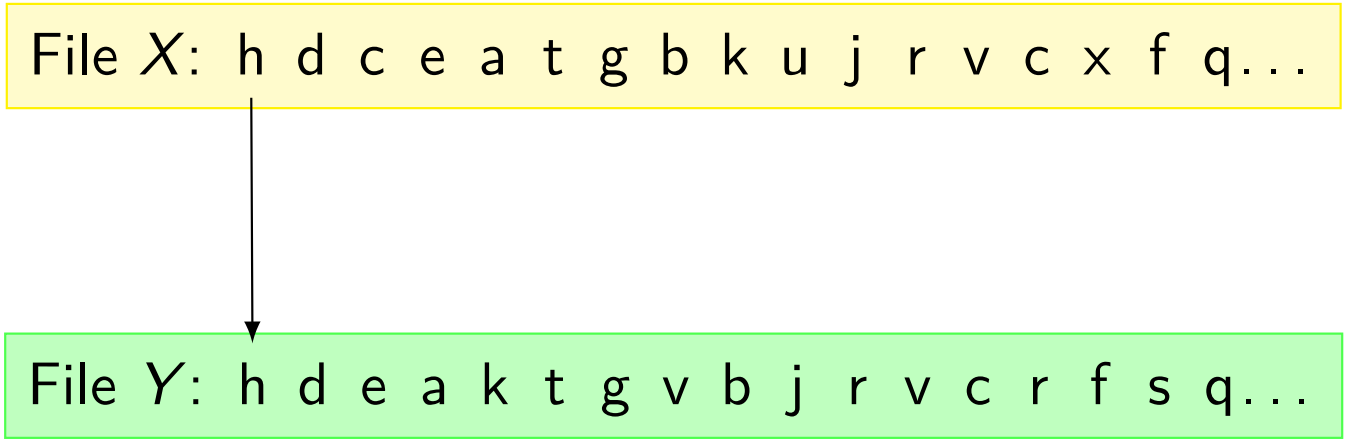
Problem Setting

File X: h d c e a t g b k u j r v c x f q...

File Y: h d e a k t g v b j r v c r f s q...

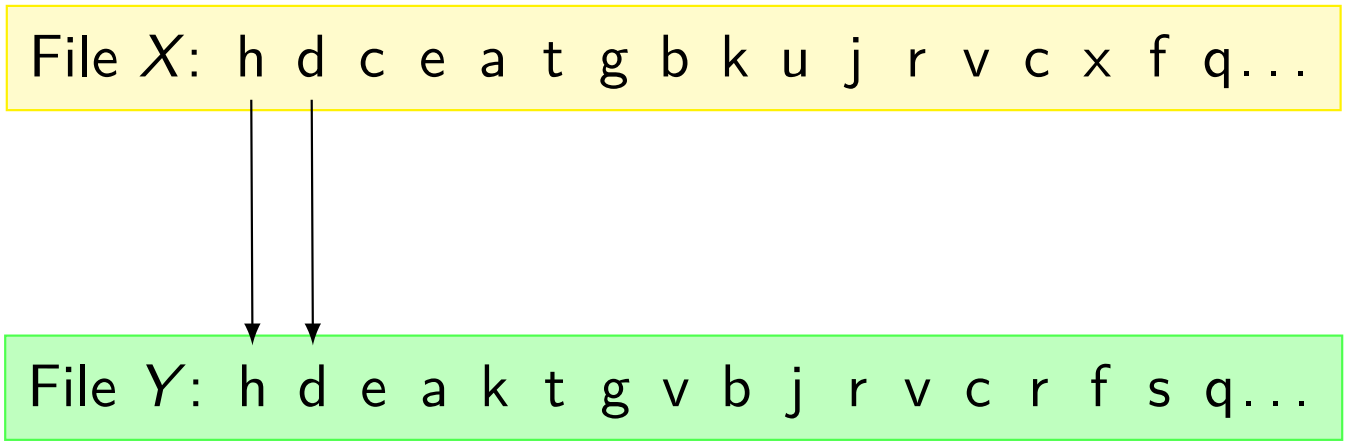
Small rate of edits:
Each symbol is edited with probability $\beta \ll 1$.

Problem Setting



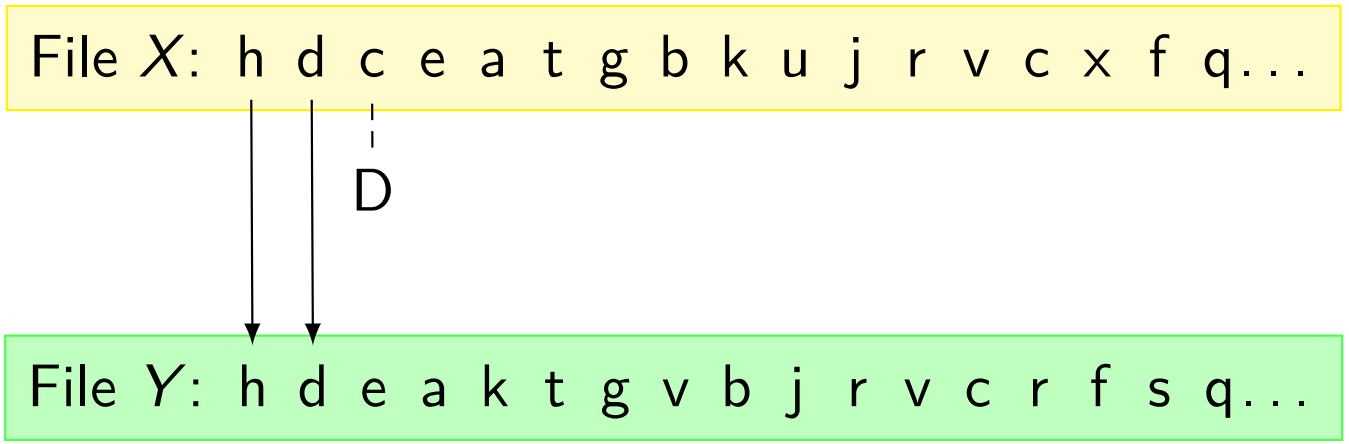
Small rate of edits:
Each symbol is edited with probability $\beta \ll 1$.

Problem Setting



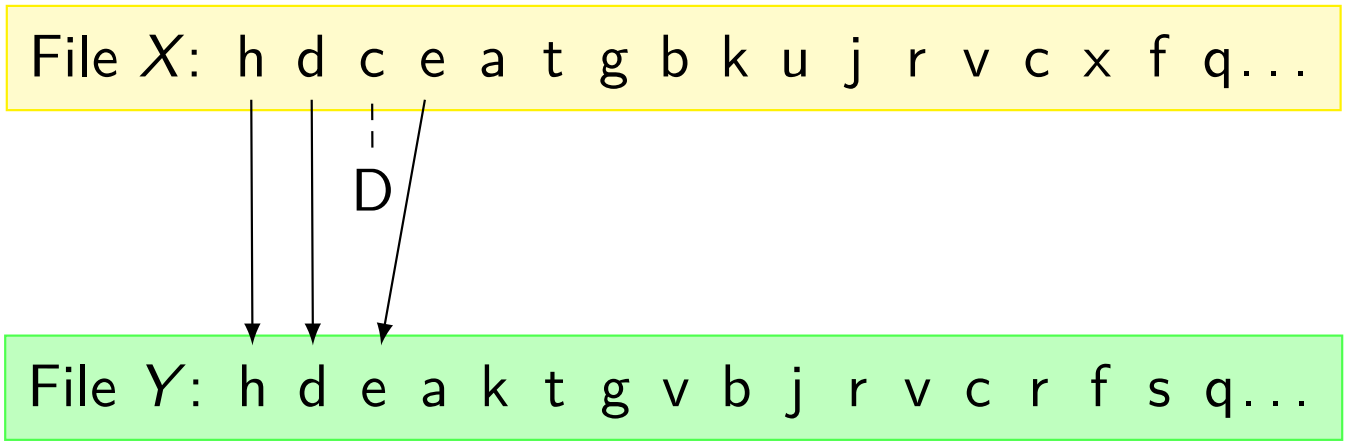
Small rate of edits:
Each symbol is edited with probability $\beta \ll 1$.

Problem Setting



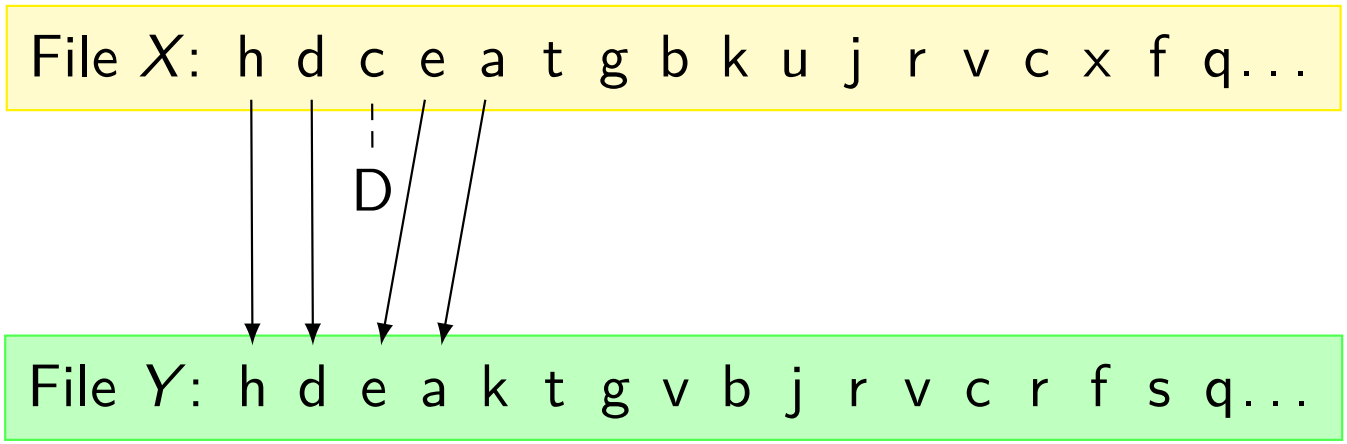
Small rate of edits:
Each symbol is edited with probability $\beta \ll 1$.

Problem Setting



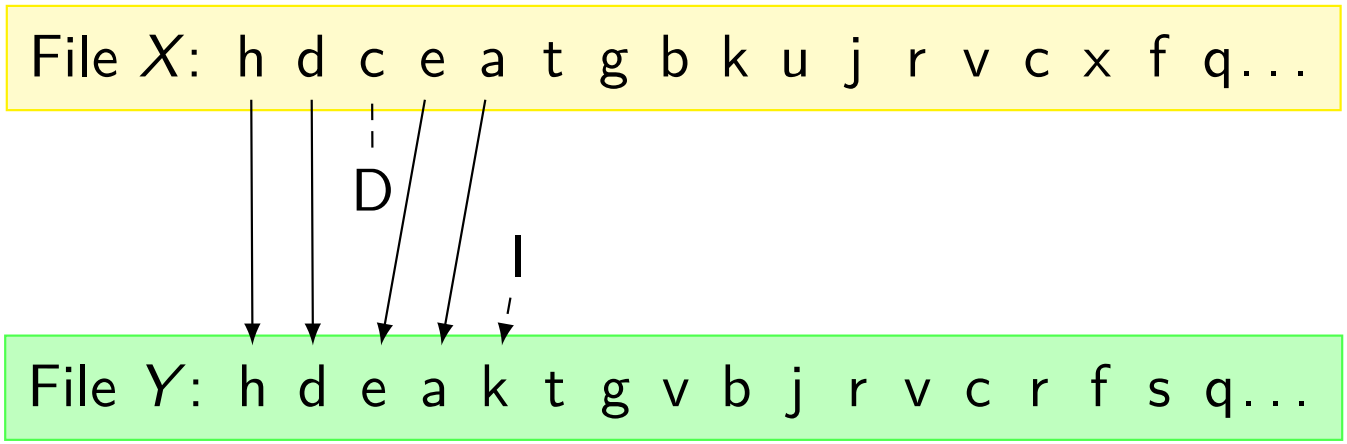
Small rate of edits:
Each symbol is edited with probability $\beta \ll 1$.

Problem Setting



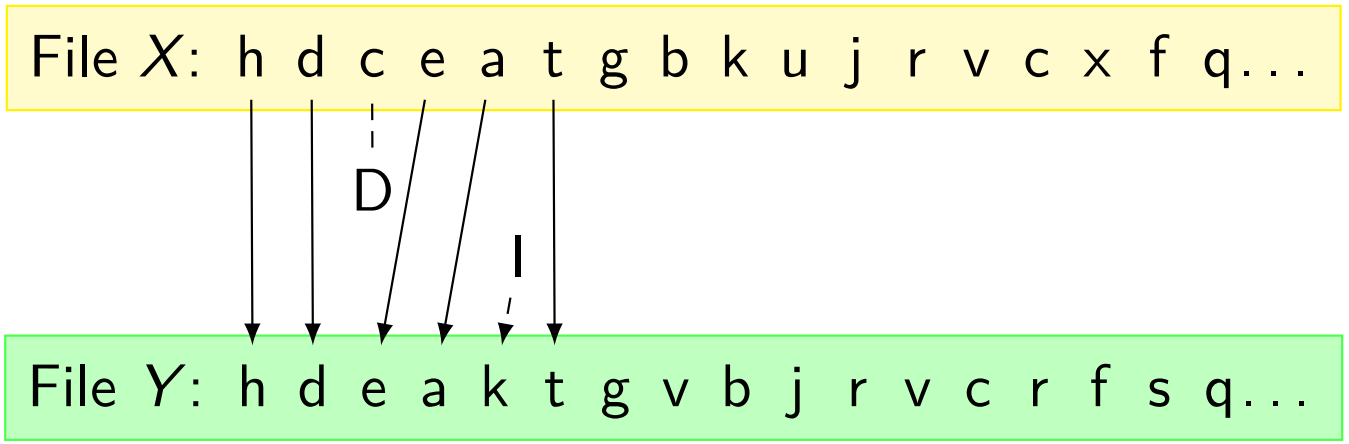
Small rate of edits:
Each symbol is edited with probability $\beta \ll 1$.

Problem Setting



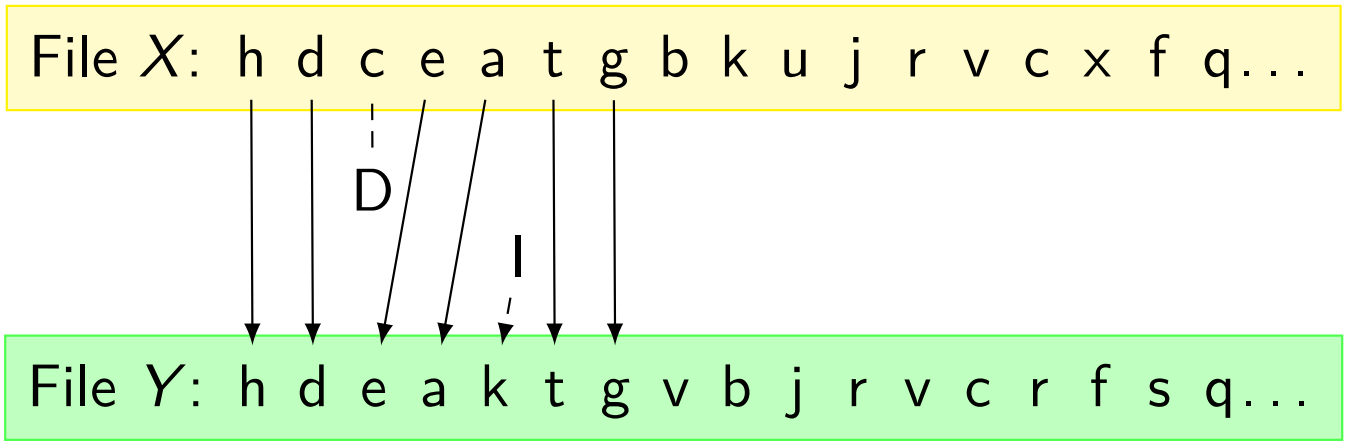
Small rate of edits:
Each symbol is edited with probability $\beta \ll 1$.

Problem Setting



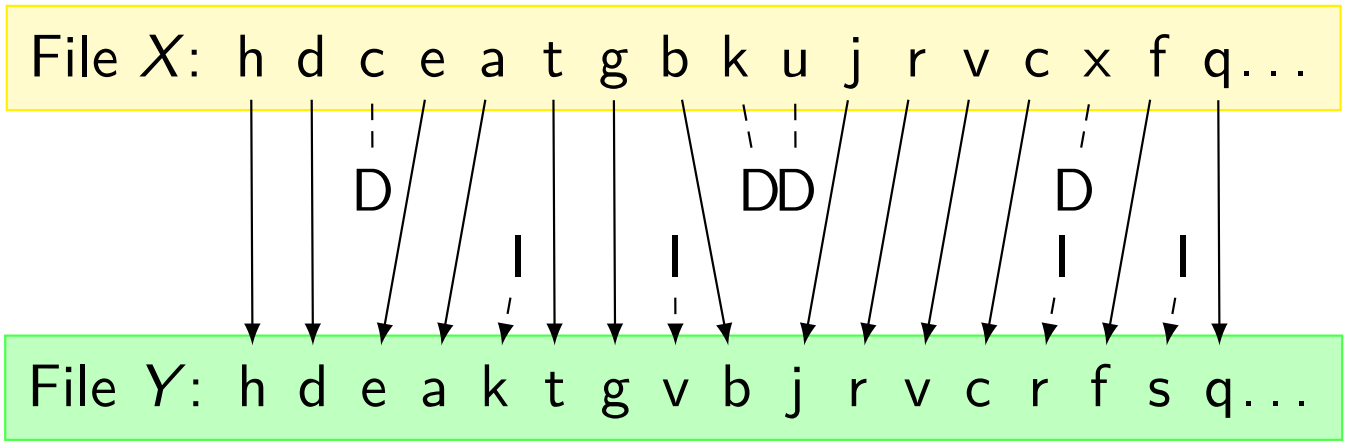
Small rate of edits:
Each symbol is edited with probability $\beta \ll 1$.

Problem Setting



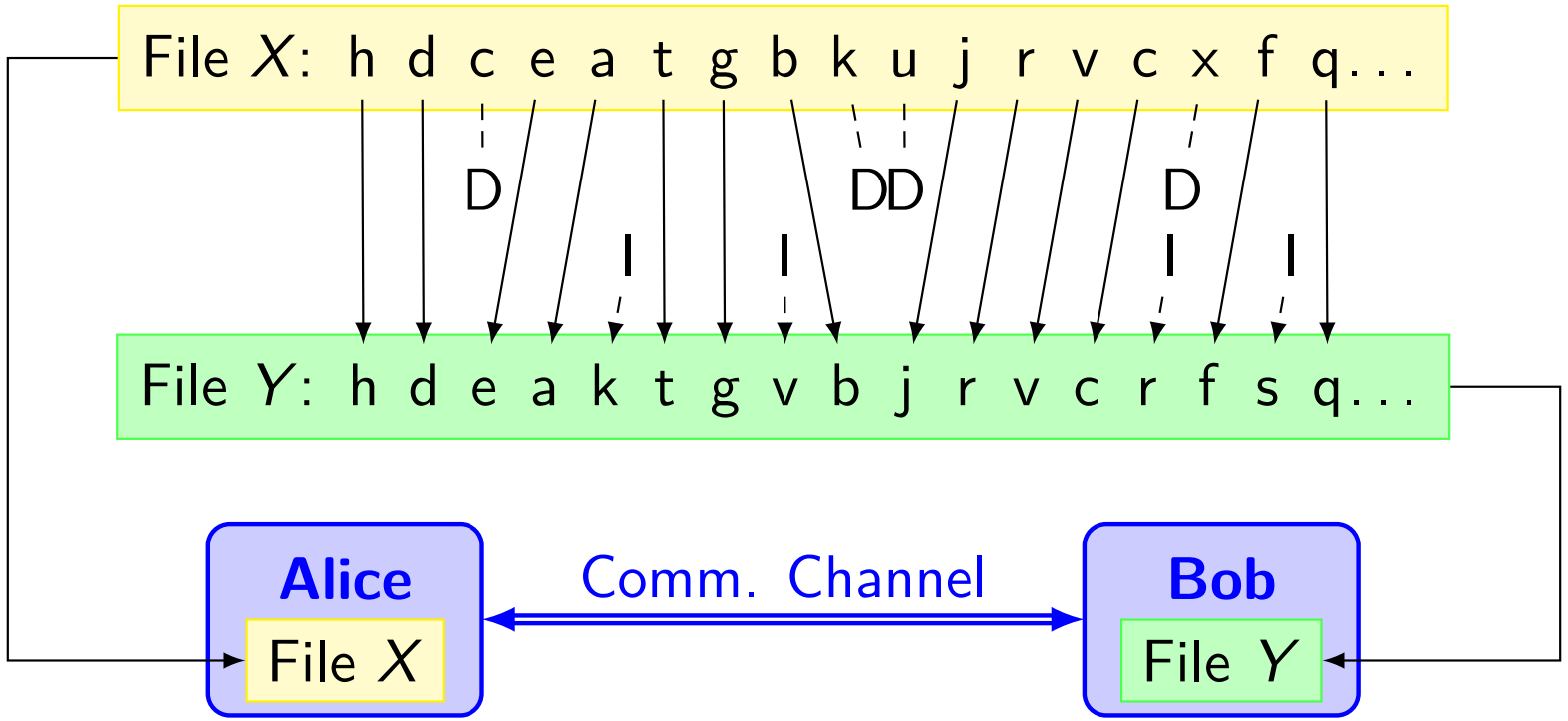
Small rate of edits:
Each symbol is edited with probability $\beta \ll 1$.

Problem Setting



Small rate of edits:
Each symbol is edited with probability $\beta \ll 1$.

Problem Setting



Goal: Interactive Communication Scheme

Allow Bob to recover X from Y:

- with low probability of error,
- with low communication cost.

Prior Related Work

- Scheme that corrects a single edit (non-binary):

G. M. Tenengolts, “Nonbinary codes, correcting single deletion or insertion”, 1984.

Prior Related Work

- Scheme that corrects a single edit (non-binary):

  G. M. Tenengolts, "Nonbinary codes, correcting single deletion or insertion", 1984.

- Scheme that corrects a fixed number of edits (binary):

  R. Venkataramanan, H. Zhang, and K. Ramchandran, "Interactive low-complexity codes for synchronization from deletions and insertions", 2012.

Prior Related Work

- Scheme that corrects a single edit (non-binary):

  G. M. Tenengolts, "Nonbinary codes, correcting single deletion or insertion", 1984.

- Scheme that corrects a fixed number of edits (binary):

  R. Venkataramanan, H. Zhang, and K. Ramchandran, "Interactive low-complexity codes for synchronization from deletions and insertions", 2012.

- Theoretical bound for the fixed rate of edits case:

  N. Ma, K. Ramchandaran and D. Tse, "Efficient file synchronization: a distributed source coding approach", 2011.

Prior Related Work

- Scheme that corrects a single edit (non-binary):

  
  G. M. Tenengolts, "Nonbinary codes, correcting single deletion or insertion", 1984.

- Scheme that corrects a fixed number of edits (binary):

  
  R. Venkataramanan, H. Zhang, and K. Ramchandran, "Interactive low-complexity codes for synchronization from deletions and insertions", 2012.

- Theoretical bound for the fixed rate of edits case:

  
  N. Ma, K. Ramchandaran and D. Tse, "Efficient file synchronization: a distributed source coding approach", 2011.

- Our work: scheme for fixed rate of edits:

  
  N. Bitouzé and L. Dolecek, "Synchronization from insertions and deletions under a non-binary, non-uniform source", 2013.

  
  S. M. S. Tabatabaei and L. Dolecek, "A deterministic, polynomial-time protocol for synchronizing from deletions", 2013.

Overview of our Synchronization Scheme

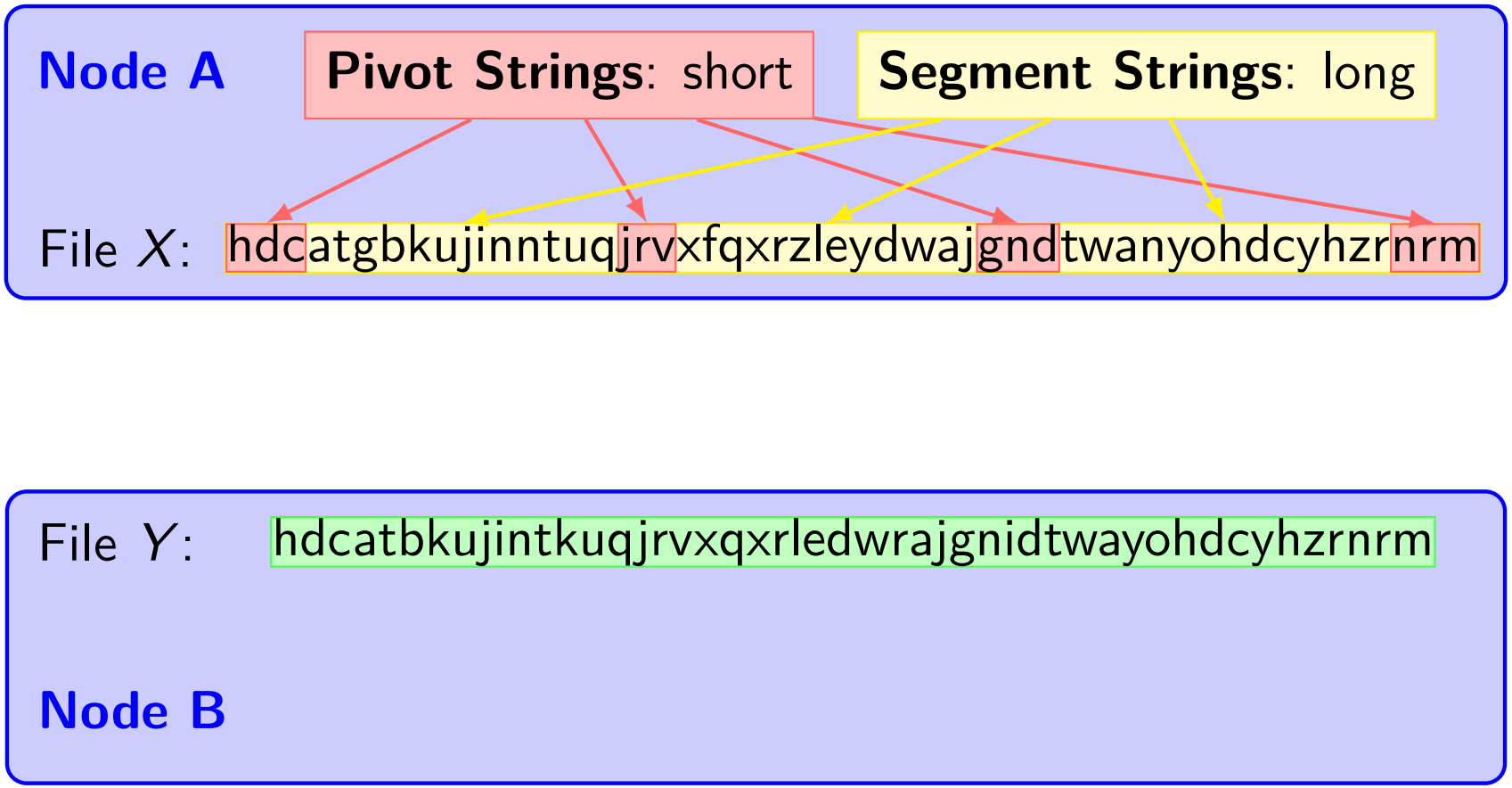
Node A

File X: hdcatgbkujinntuqjrvxfqxrzleydwajgndtwanyohdcyhzrnrm

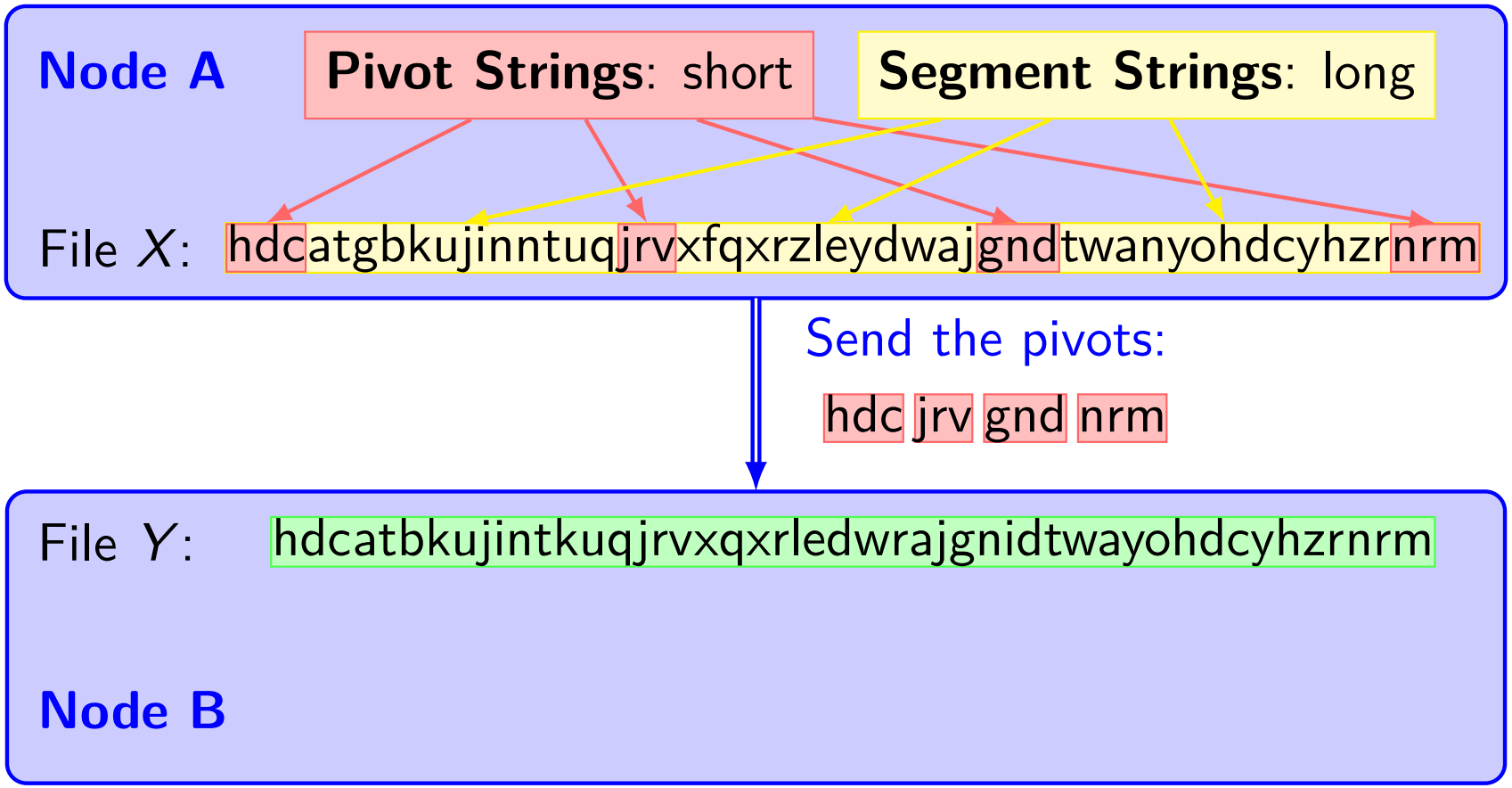
File Y: hdcatbkujintkuqjrvxqxrledwrajgnidtwayohdcyhzrnrm

Node B

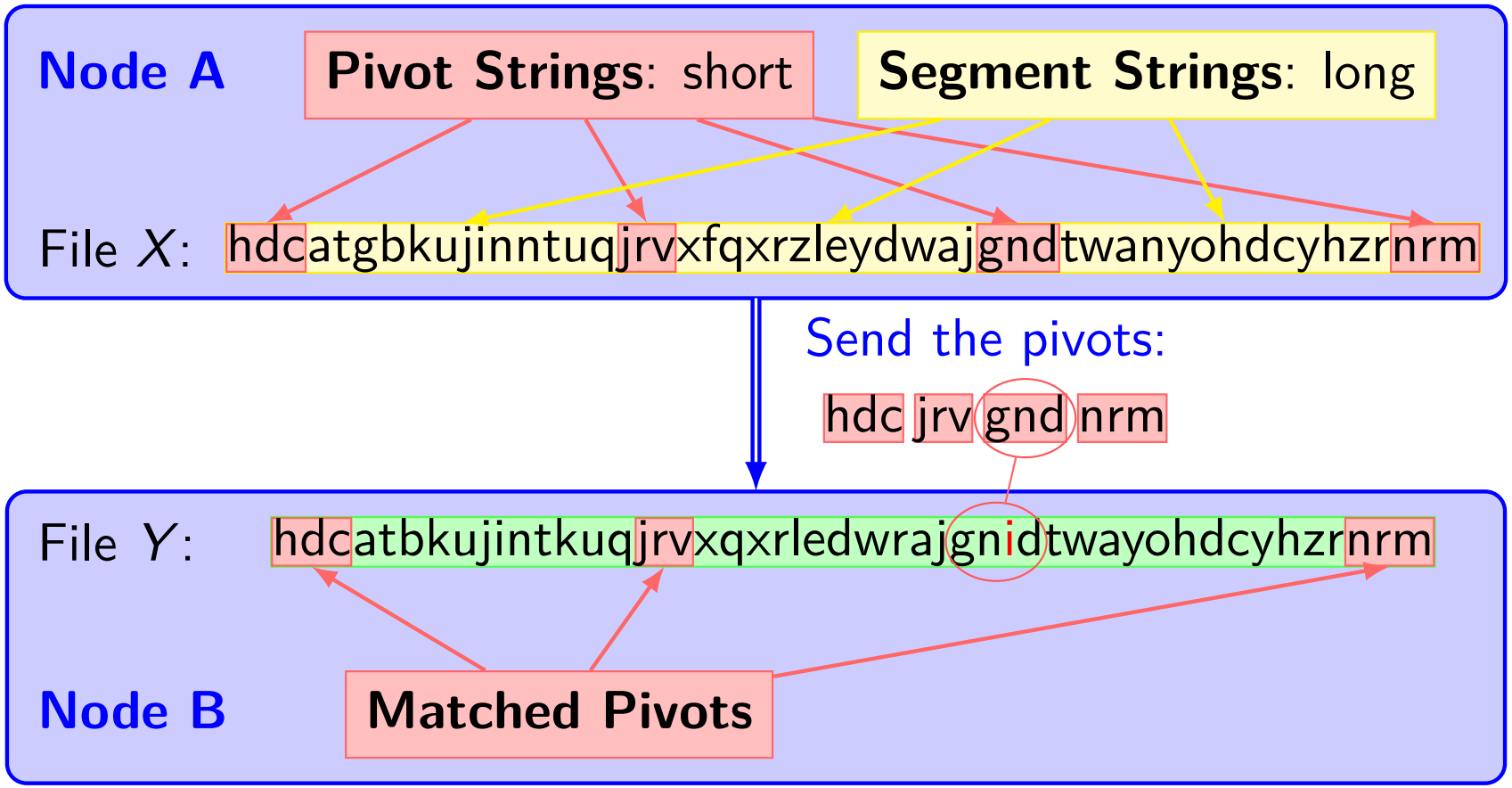
Overview of our Synchronization Scheme



Overview of our Synchronization Scheme

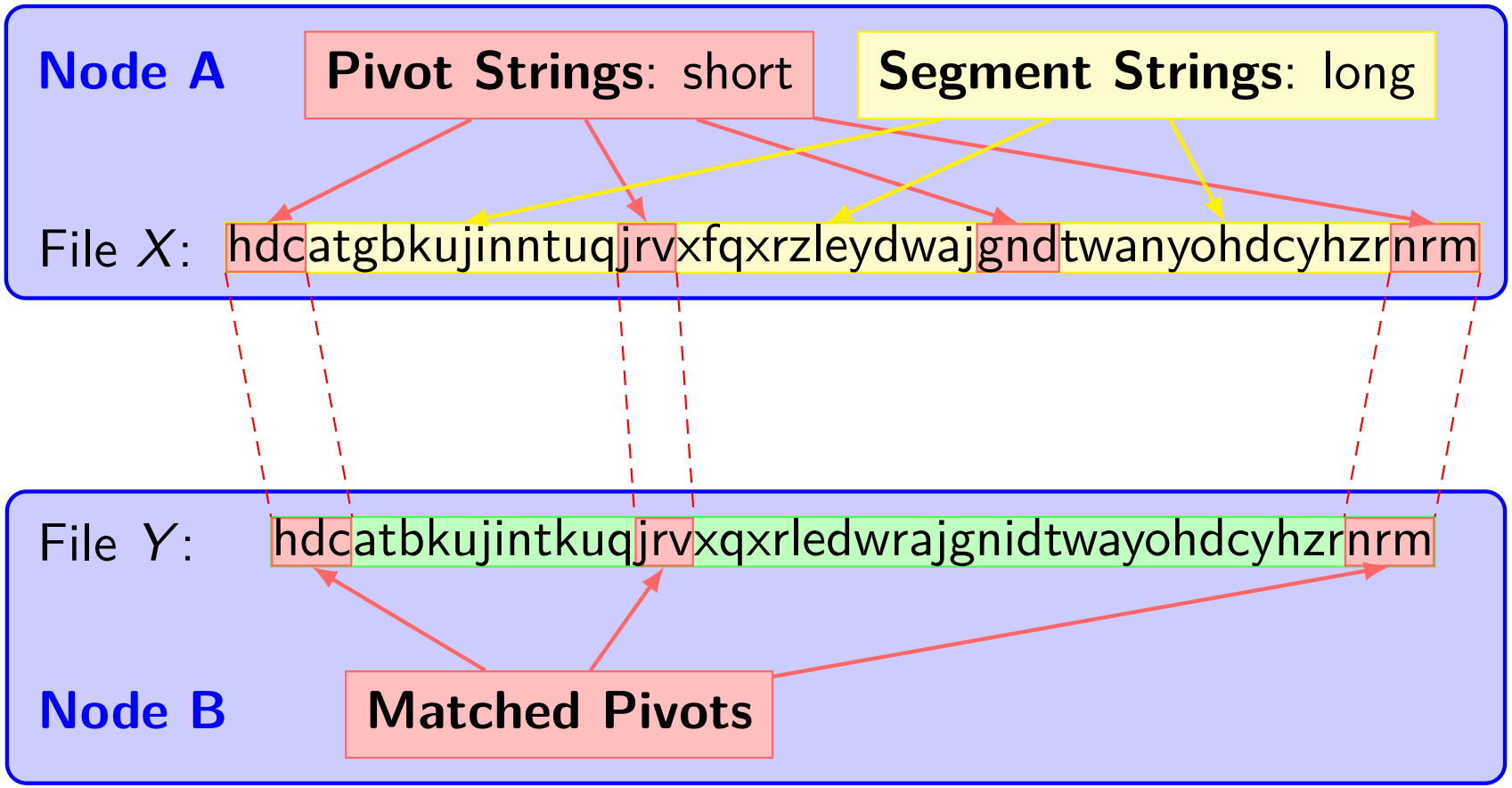


Overview of our Synchronization Scheme



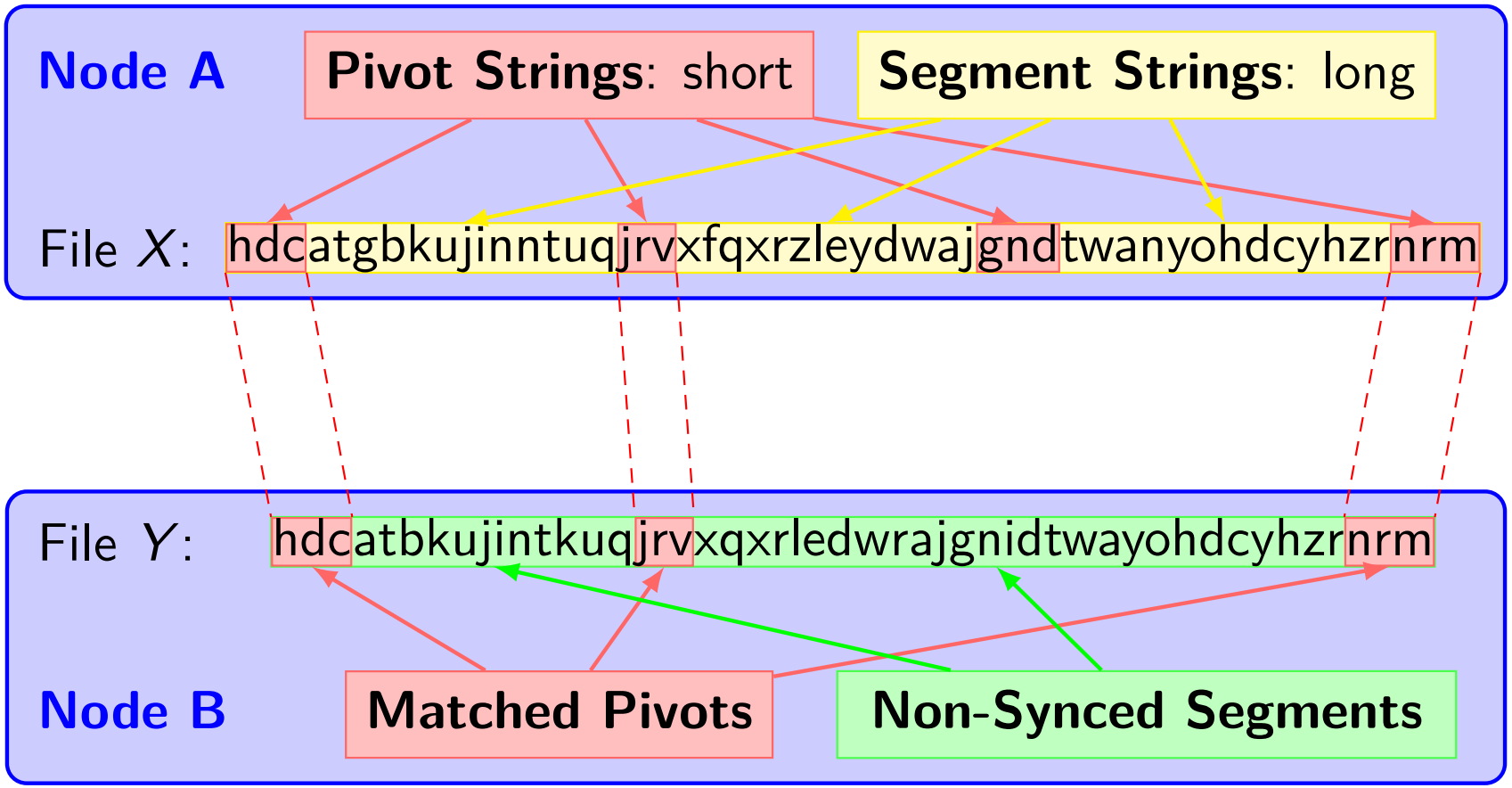
1. Matching Module: Matches the pivot strings.

Overview of our Synchronization Scheme



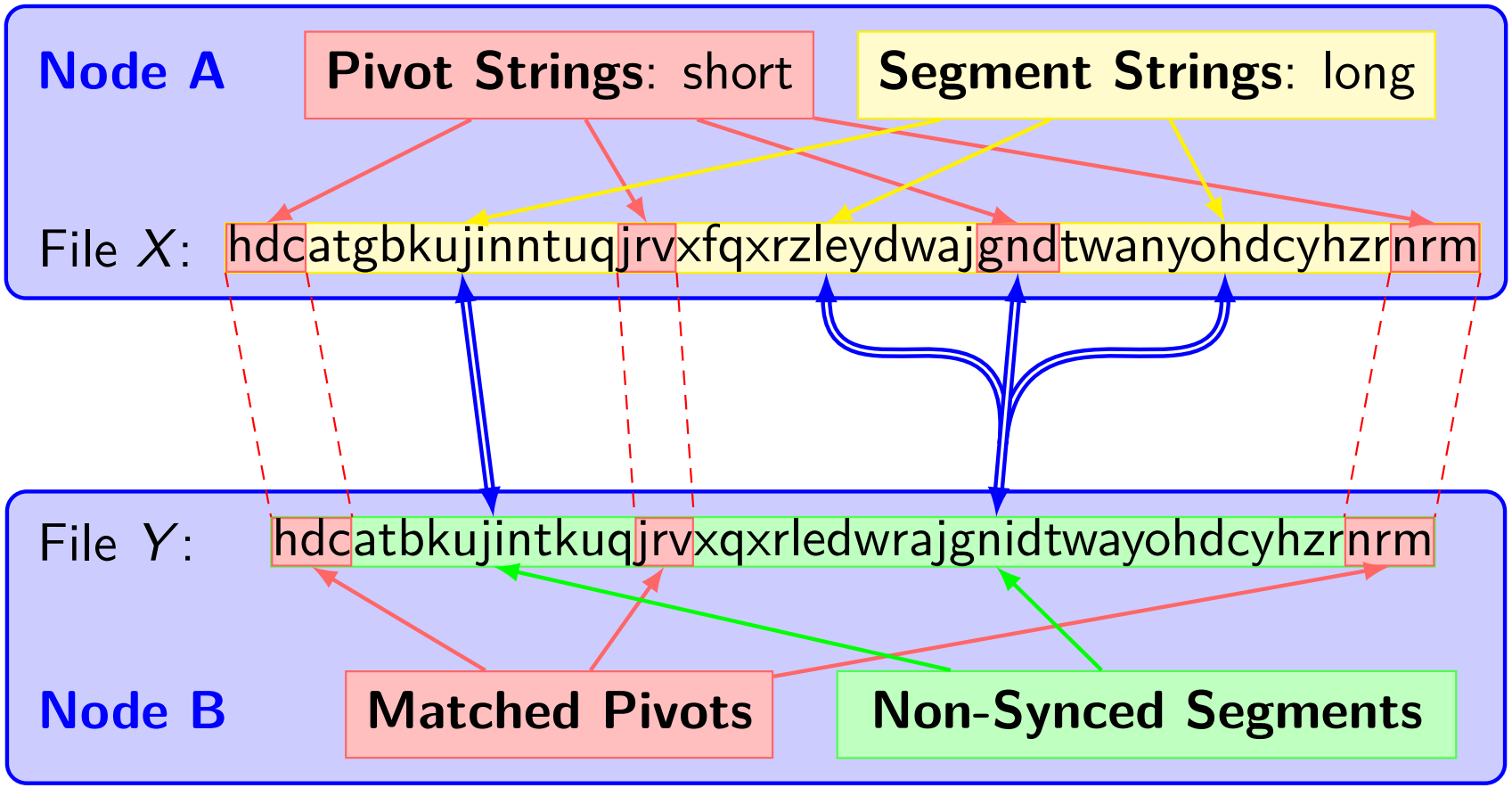
1. Matching Module: Matches the pivot strings.

Overview of our Synchronization Scheme



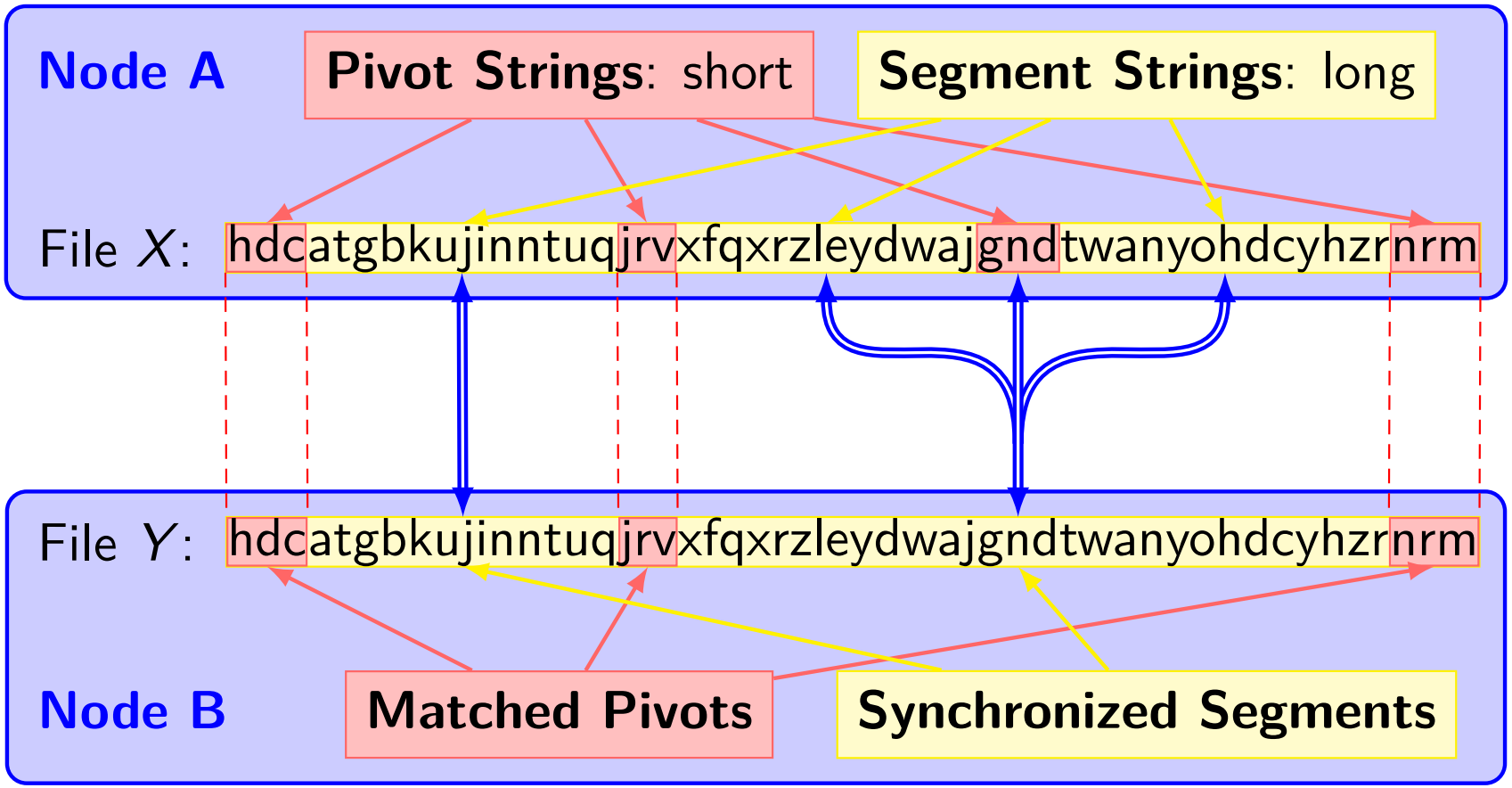
1. Matching Module: Matches the pivot strings.

Overview of our Synchronization Scheme



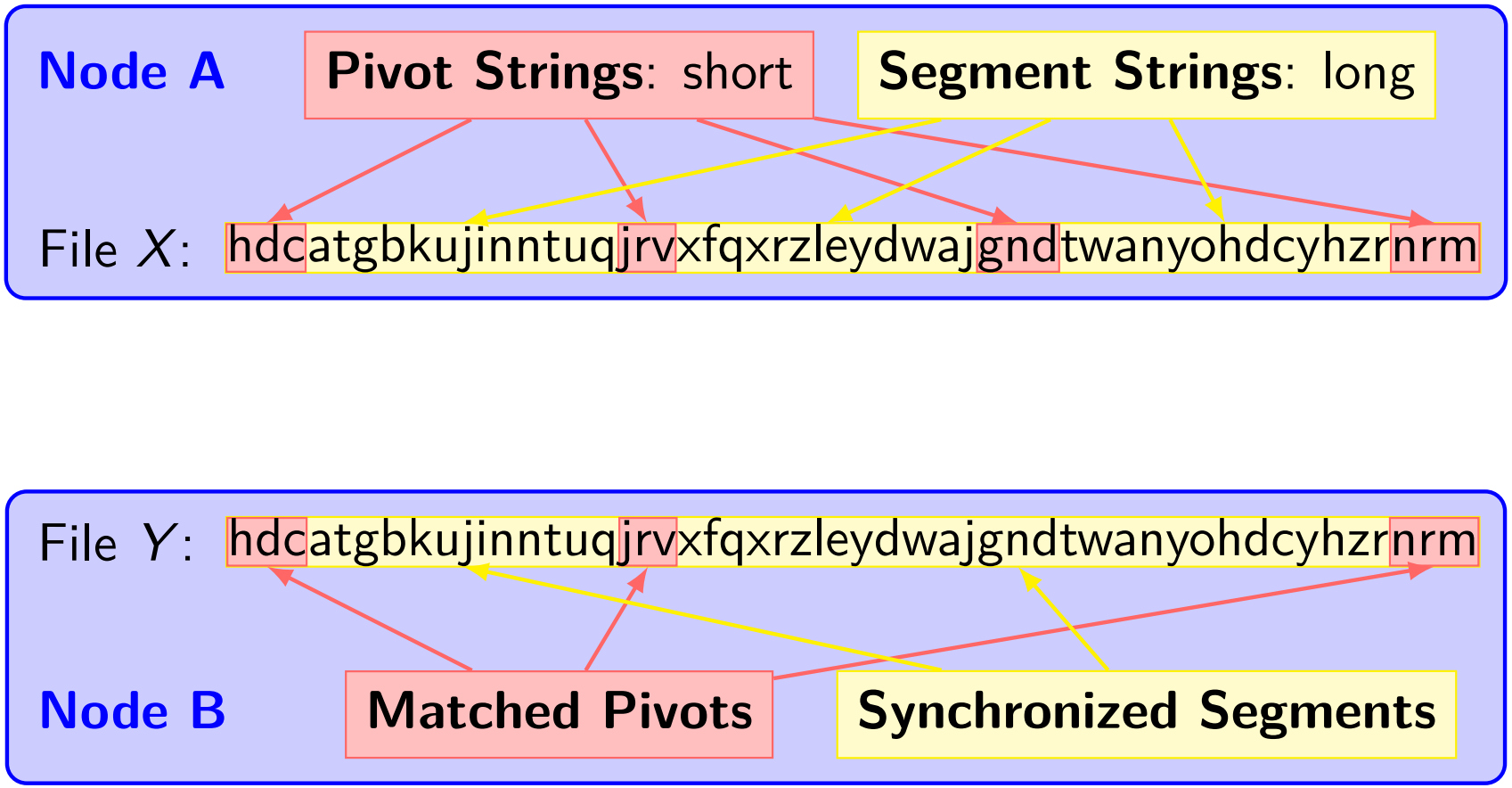
1. Matching Module: Matches the pivot strings.
2. Edit Recovery Module: Synchronizes the segment strings.

Overview of our Synchronization Scheme



1. Matching Module: Matches the pivot strings.
2. Edit Recovery Module: Synchronizes the segment strings.

Overview of our Synchronization Scheme



1. Matching Module: Matches the pivot strings.
2. Edit Recovery Module: Synchronizes the segment strings.
3. Channel Coding Module: Recovers from residual errors if any.

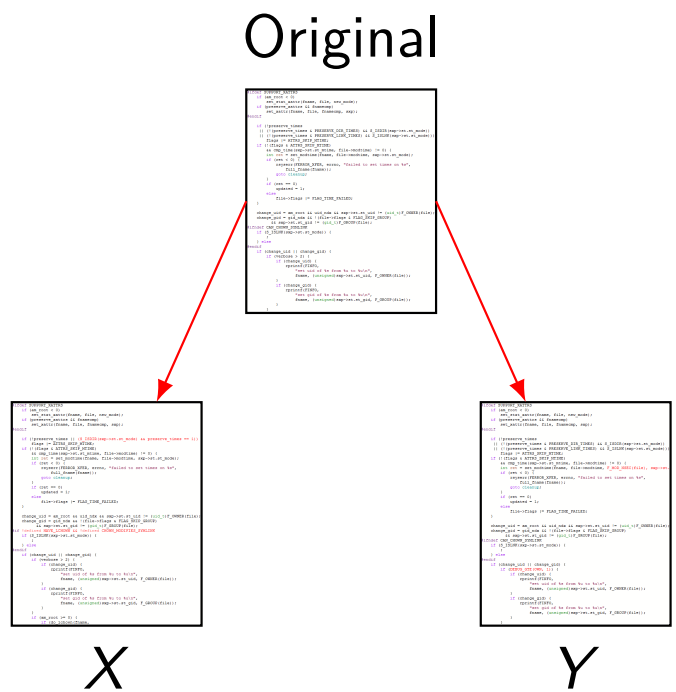
Main Result

Considering the scenario where

- File Y is an edited version of file X,
- Alice has file X,
- Bob has file Y,
- Alice $\overset{\text{Comm. Channel}}{\longleftrightarrow}$ Bob,

we provide a synchronization scheme with

- little communication required (within a constant factor of optimal rate),
- negligible error probability.



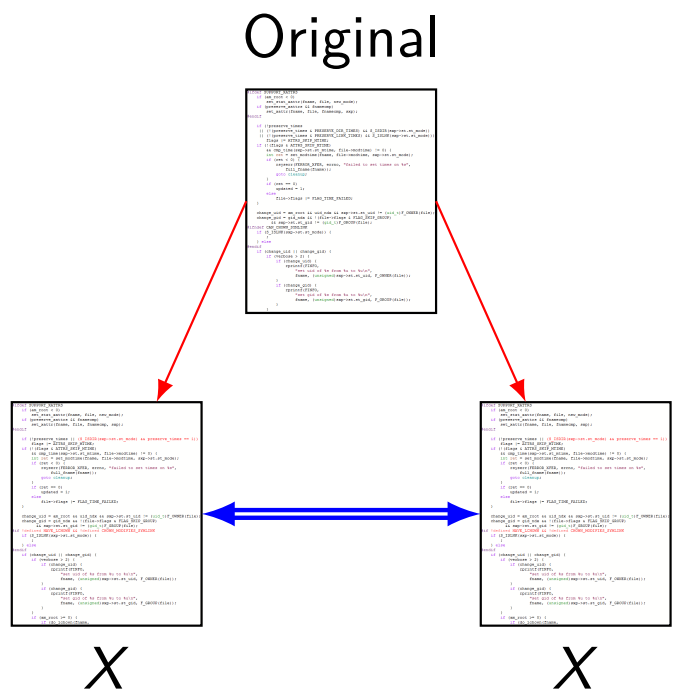
Main Result

Considering the scenario where

- File Y is an edited version of file X,
- Alice has file X,
- Bob has file Y,
- Alice $\overset{\text{Comm. Channel}}{\longleftrightarrow}$ Bob,

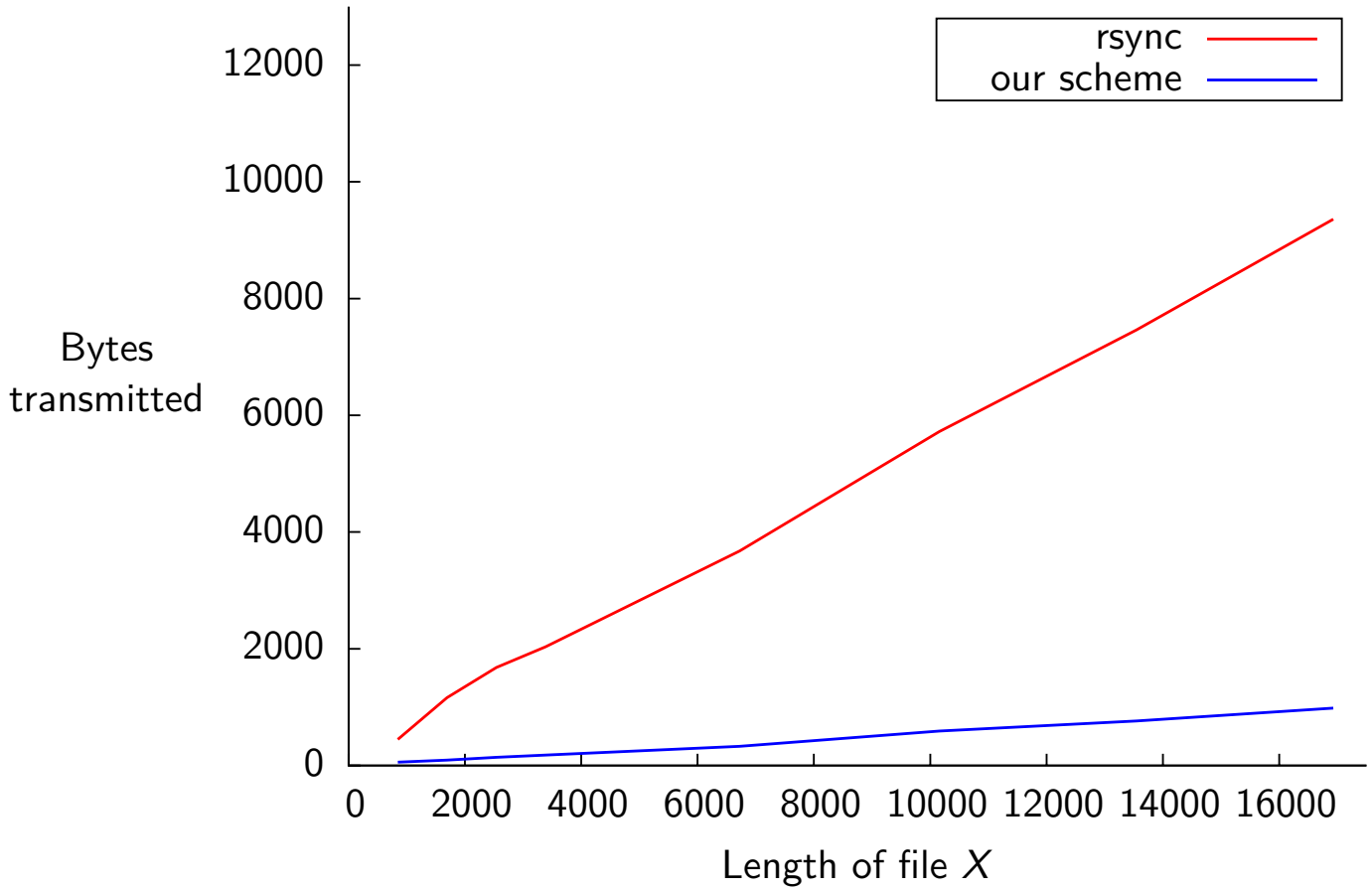
we provide a synchronization scheme with

- little communication required (within a constant factor of optimal rate),
- negligible error probability.



Comparison with rsync

- $\beta = 0.01$, i.i.d. file, i.i.d. edits,
- rsync block-size optimized,
- Our pivot length: 5, segment length: 100.



Ongoing Work

- Allow for more complex edit patterns
  (e.g., in practical scenarios, edits are often “bursty”),

Ongoing Work

- Allow for more complex edit patterns
  (e.g., in practical scenarios, edits are often “bursty”),
- Specialize our scheme to application-dependent types of files
  (e.g., if the files are source code, exploit that structure),

Ongoing Work

- Allow for more complex edit patterns
  (e.g., in practical scenarios, edits are often “bursty”),
- Specialize our scheme to application-dependent types of files
  (e.g., if the files are source code, exploit that structure),
- Optimize the implementation
  (in terms of both computation and network usage).