平成13年度 情報科学セミナー 資料

Gnutella Protocol 概要

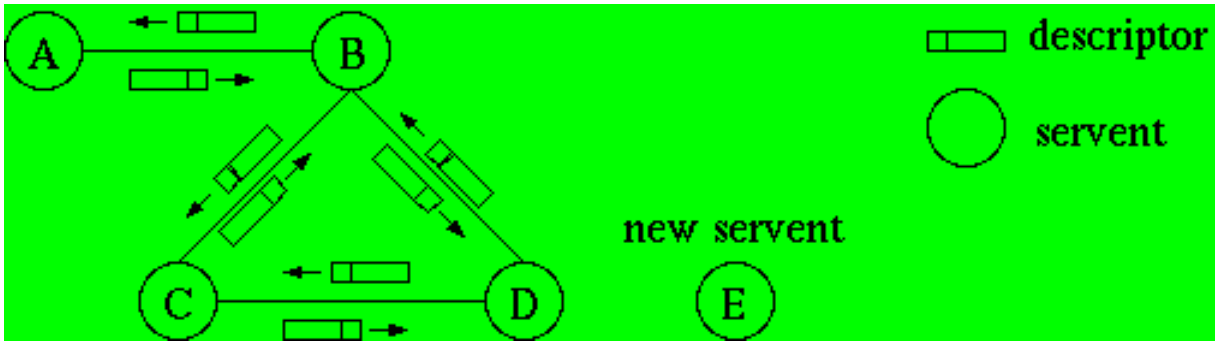
算法設計工学研究室
丸山 一貴
kazutaka@sanpo.t.u-tokyo.ac.jp

Gnutella Network

□ファイル共有のためのネットワーク
- ○多数のノード（servent）が接続

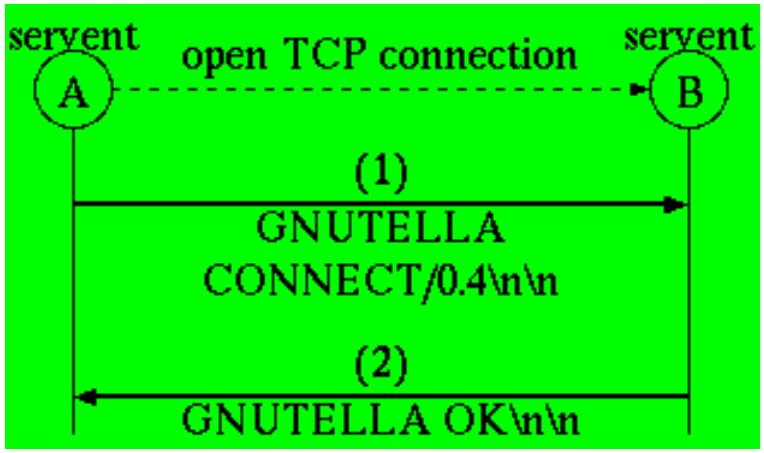
□メッセージ（descriptor）の交換で実現
- ○servent 追加、ファイルの検索、...

□descriptor をルーティング
- ○隣接する servent に送信
- ○受信した servent は他の servent にフォワード



ネットワークへの接続手順

- 「隣の」servent に TCP で接続
  - gnut> open host:port
- 「GNUTELLA CONNECT/0.4\n\n」を送信
  - こちらが Gnutella servent であることを通知
- 「GNUTELLA OK\n\n」を受信
  - 相手も Gnutella servent であることを確認



Descriptor の構造と種類

□ヘッダとペイロード
- ○ヘッダは固定長 (23バイト)
- ○ペイロードは可変長

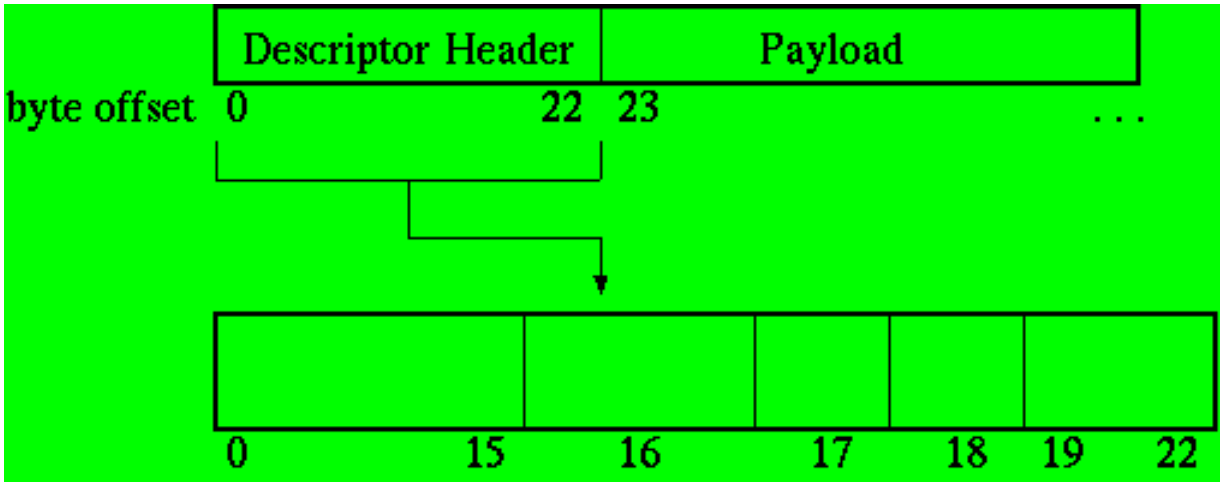
□5種類の descriptor
- ○Ping, Pong：自分の存在の通知・応答
- ○Query, QueryHit：ファイルの検索・応答
- ○Push：ファイルのプッシュ型転送用

□エラー修正はしない
- ○伝送中のエラーは TCP 以下のレイヤーで対応済み
- ○不正な descriptor を送ってきた回線は切断

□network byte order (big endian)

Descriptor Header の構造

- □5つのフィールド
  - ○Descriptor ID：descriptor に固有の番号
  - ○Payload Descriptor：Payload の種類
  - ○TTL：Time-To-Live、servent 通過時に decrement
    - ▹0 になったら廃棄
  - ○Hops：servent 経由段数、servent 通過時に increment
  - ○Payload Length：ペイロード長
    - ▹descriptor 間には gap/padding なし
    - ▹preemble などの同期手順なし



Descriptor（１）：Ping

- Ping（0x00）
  - ペイロードはなし（Payload Length = 0）
  - 能動的に他の servent を探すときに使用
    - Ping を受け取ったら Pong を返す
  - 送信頻度については規定なし

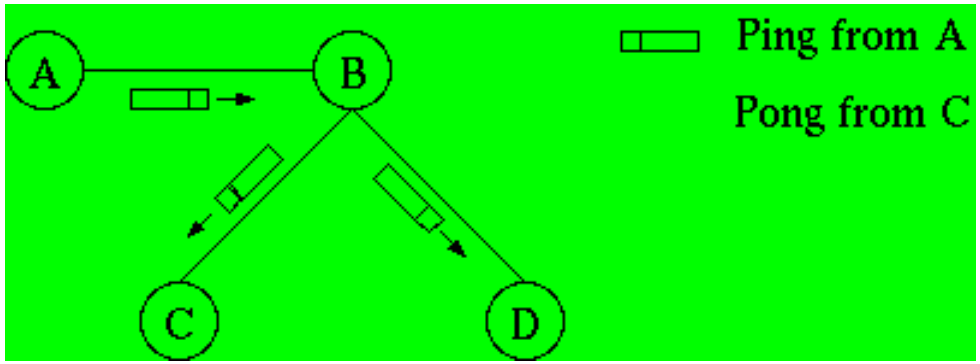
Descriptor（２）：Pong

- Pong（0x01）
  - Ping に対する応答

- ペイロードの構造
  - Port：送信者のポート番号
  - IP Addr.：送信者の IP アドレス
    - これは little endian
  - Num. of Files Shared：公開ファイルの数
  - Num. of Kilobytes Shared：公開ファイルサイズの合計

Descriptor Routing - Part 1

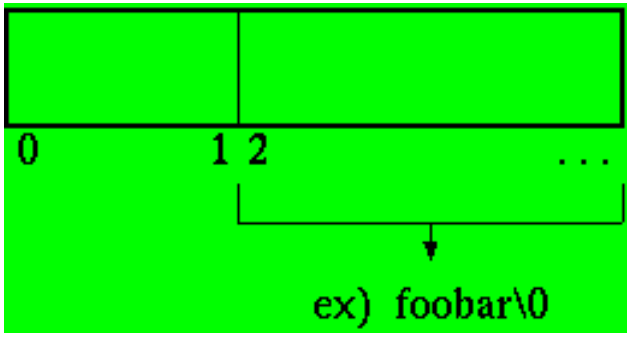
- Ping は直接接続している全ての servent にフォワード (送り主は除く)
- Ping/Pong は同ルート
  - Descriptor ID = n である Pong が来たときに、DID = n のPing を通した記録がなければ、その Pong を廃棄
  - ある Ping に対する Pong は、同じ DID を用いる
- フォワードした descriptor を記録すれば良い
  - {DID, incoming_socket} で管理できる



Descriptor（３）：Query

- Query（0x80）
  - ファイルの検索に使用

- ペイロードの構造
  - Minimum Speed：検索応答時の接続回線速度の最小値
    - 設定されている接続回線速度（56k, T1, ...）値がこれより小さかったら、検索にヒットしても応答しない
  - Search Criteria：検索キーワード
    - null terminate された文字列
    - gnut は3文字以下の場合に応答しない？



Descriptor（４）：QueryHit（1/2）

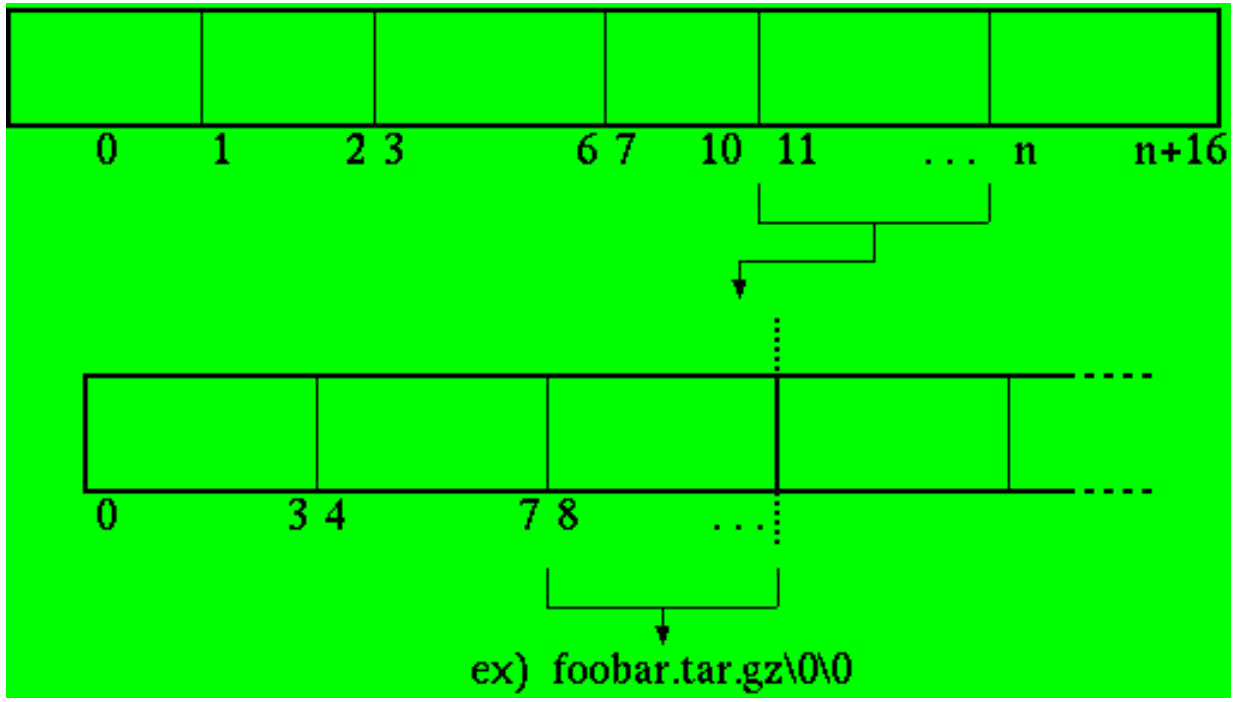
- □QueryHit（0x81）
  - ○検索にヒットしたときの応答

- □ペイロードの構造
  - ○Num. of Hits：検索ヒット数
  - ○Port：送信者のポート番号
  - ○IP Addr.：送信者の IP アドレス
  - ○Speed：送信者の接続回線速度
  - ○Result Set：検索結果群
    - ▹Num. of Hits の分だけ検索結果が入っている
  - ○Servent Identifier：Push 型転送要求時に使用

Descriptor（４）：QueryHit（2/2）

□Result Set の構造

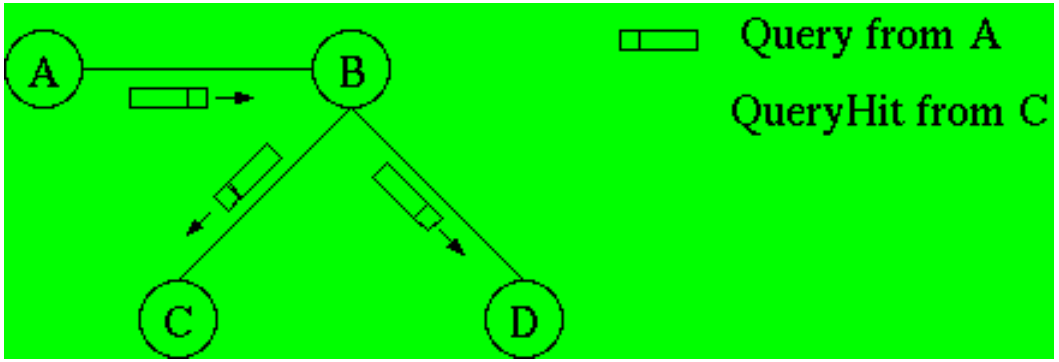
- ○File Index：送信者のファイル管理番号
- ○File Size：ファイルサイズ
- ○File Name：ファイル名
  - ▹double null terminate された文字列
  - ▹"\0\0" が次の Result との delimiter として機能



Descriptor Routing - Part 2

- Query は直接接続している全ての servent にフォワード (送り主は除く)

- Query/QueryHit は同ルート
  - Ping/Pong 同様、DID でルーティング



ファイルの転送（１）

□HTTP を用いて直接通信

○ファイル要求 servent（A）が HTTP クライアント

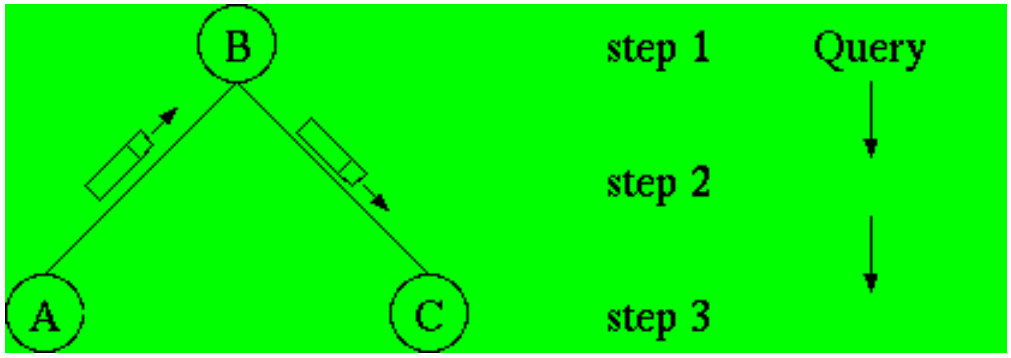
▹GET /get/<FileIndex>/<FileName> HTTP/1.0を送信

○ファイル保持 servent（C）が HTTP サーバ

▹HTTP 200 OKで応答

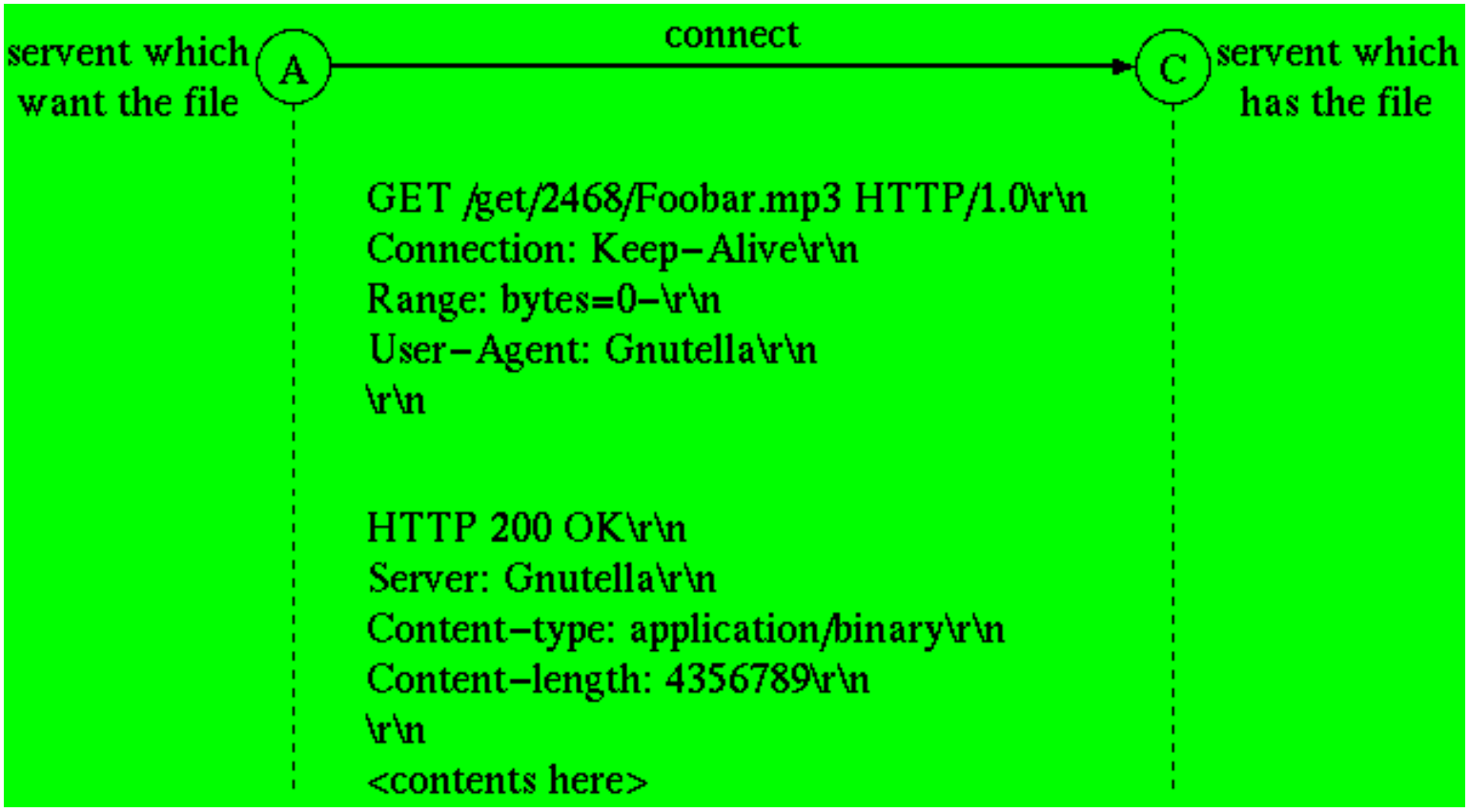
○HTTP の Range フィールドをサポート

▹転送が途切れても、残りだけを送り直せる



ファイルの転送（２）

- □QueryHit の Result Set が以下のようなとき
  - ○File Index：2468
  - ○File Size：4356789
  - ○File Name：Foobar.mp3\0\0



Descriptor（５）：Push（1/3）

□Push（0x40）

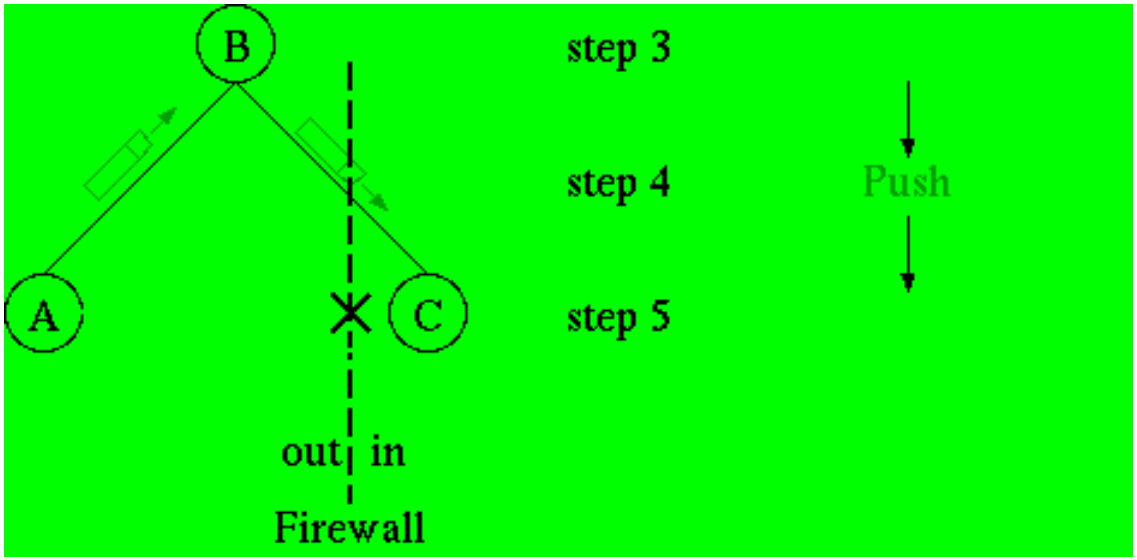
○プッシュ型転送要求メッセージ

□ファイル保持 servent（Ｃ）が firewall の内側のとき...

○要求 servent（Ａ）の HTTP connection は失敗

○Push descriptor を、保持 servent（Ｃ）の方へ送信

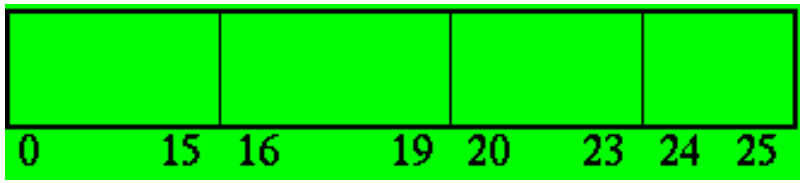
○保持 servent（Ｃ）から要求 servent（Ａ）へ（逆向きに）HTTP connection



Descriptor（５）：Push（2/3）

□ペイロードの構造

- ○Servent Identifier：ファイル保持 servent 識別子
  - ▹QueryHit descriptor で通知済み
  - ▹ファイル保持 servent が、自分宛であることを知るために必要
- ○File Index：ファイル保持 servent 上でのファイル識別子
  - ▹QueryHit descriptor で通知済み
  - ▹このファイルを送ってもらう
- ○IP Addr.：ファイル要求 servent の IP アドレス
  - ▹このアドレスに接続してもらう
- ○Port：ファイル要求 servent のポート番号

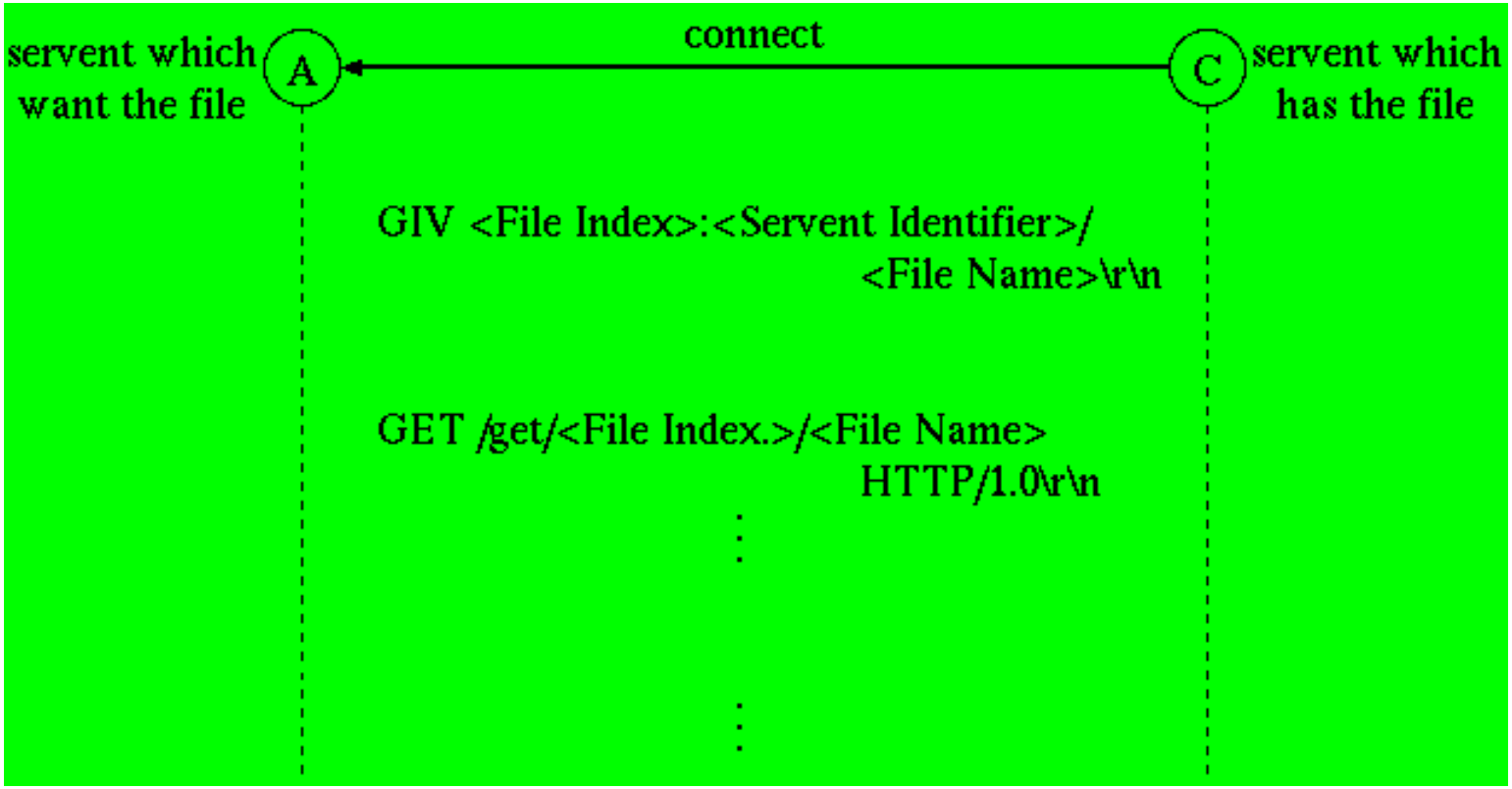


Descriptor（５）：Push（3/3）

□保持 servent（Ｃ）から内容を通知

○GIV <File Index>:<Servent Identifier>/<File Name>\r\n

□要求 servent（Ａ）が通常の手順で request



Descriptor Routing - Part 3

- Push は QueryHit と同ルート
  - Servent ID = n である Push が来たときに、SID = n のQueryHit を通した記録がなければ、その Push を廃棄
  - Push は DID ではなく SID でルーティングされる
  - Push は QueryHit とは異なる (新しい) DID をつける
- DID と Payload Descriptor が、見たことあるものだったらフォワードしない