

1. Сверточные коды. Кодирование, декодирование в канале со стираниями.
2. Протокольные последовательности (мы начнем с генерации случайных последовательностей заданного веса и затем рассмотрим более интересные классы последовательностей с малым числом пересечений. Идея использования протокольных последовательностей с малым числом пересечений с последующей коррекцией потерянных пакетов с помощью корректирующих кодов изложена в Massey.pdf). Недостаток системы, рассмотренной в этой работе – слишком длинные блоковые коды, что приводит к большой задержке. Мы надеемся уменьшить задержку, используя сверточные коды.
3. Наша цель – смоделировать следующую систему:

М пользователей генерируют свои пакеты (в нашей системе пользователь посылает сообщение с заданной вероятностью  $p_i$ ,  $i=1,\dots,M$ , с вероятностью  $(1-p_i)$  пользователь “молчит”). Если пользователь имеет сообщение для передачи, он случайно выбирает протокольную последовательность из списка разрешенных протокольных последовательностей.

После того как выбрана протокольная последовательность, пользователь передает пакеты на “единицах” этой последовательности и молчит на ее “нулях”.

Мы начнем с этой простой стратегии. Далее мы рассмотрим разные стратегии, основанные на наблюдении пользователями ситуации в канале.

Сообщение пользователя состоит из посылок по  $N$  пакетов каждая, первые  $K$  пакетов из каждого  $N$ - информационные, следующие за ними  $N$  пакетов служат для коррекции потери пакетов. Мы используем сверточный код со скоростью  $K/N$ , т.е.  $N$  соответствует одному ребру в решетке сверточного кода. Особенность сверточного кода состоит в том, что в канале со стираниями (потерями пакетов) он позволяет не только гарантировано восстанавливать  $d-1$  пакетов, но восстанавливать число пакетов больше  $d-1$ , если они достаточно разнесены по длине передаваемой последовательности. Исходя из известных корректирующих способностей сверточных кодов, можно без реализации кодирования и декодирования оценить, какие конфликты между протокольными последовательностями пользователей будут исправлены и сколько пакетов в среднем смогут передать  $M$  пользователей, т.е., пропускную способность системы (смотри пример на стр. 14 статьи).

Можно менять не только протокольные последовательности, но и стратегию поведения пользователей, чтобы избежать конфликтов. Например, как в системе R-ALOHA, пользователи конфликтуют только на этапе выбора протокольных последовательностей, при дальнейшей передаче данных конфликты отсутствуют.

Можно попытаться принять во внимание потери пакетов на физическом уровне сети связи и исправлять их потери.

Думаю, что следует начать с написания программ кодирования последовательности сверточным кодом и исправления стираний на некотором окне (срок январь).

Затем рассмотреть разные классы протокольных последовательностей, начиная со случайных последовательностей заданного веса (срок февраль).

Стратегии выбора протокольных последовательностей и оценка пропускной способности системы (март-апрель).