

Выполнил(а) Балин А. А., № группы P3112, оценка                       
Фамилия И.О. студента не заполнять

<b>Название статьи/главы книги/видеолекции</b> What is ARQ (Automatic Repeat Request)?		
<b>ФИО автора статьи (или e-mail)</b> <a href="https://clck.ru/32K7wH">https://clck.ru/32K7wH</a>	<b>Дата публикации (не старше 2019 года)</b> "08" июня 2021 г.	<b>Размер статьи (от 400 слов)</b> 860
<b>Прямая полная ссылка на источник или сокращённая ссылка (bit.ly, tr.im и т.п.)</b> <a href="https://clck.ru/32K7vX">https://clck.ru/32K7vX</a>		
<b>Теги, ключевые слова или словосочетания</b> Контроль ошибок, компьютерные сети		
<b>Перечень фактов, упомянутых в статье</b> <ol style="list-style-type: none"> <li>1. ARQ – протокол контроля ошибок в двусторонней системе связей.</li> <li>2. Протокол отправляет потерянные/повреждённые пакеты данных автоматически.</li> <li>3. Отправитель получает подтверждение, если пакет данных дошёл до получателя.</li> <li>4. Если подтверждение не получено в течение определённого времени, пакет отправляется ещё раз до тех пор, пока он не будет полностью передан.</li> <li>5. ARQ имеет широкий спектр применения: банковские системы, протокол TCP, двоичный синхронный протокол связи IBM и т.д.</li> <li>6. Есть несколько способов реализации этого алгоритма: Stop and Wait, Go Back-N, Selective Repeat/Reject.</li> <li>7. Первый вариант отправляет пакеты по одному до тех пор, пока не получит подтверждения от получателя, новые пакеты при этом не отправляются.</li> <li>8. Второй вариант подразумевает, что получатель отслеживает, какой пакет он должен принять следующим, и отправляет номер пакета отправителю вместе с подтверждением текущего пакета.</li> <li>9. При этом в варианте Go Back-N отправитель высылает пакеты по несколько штук без ожидания подтверждения, в случае потери одного он возвращается к последнему подтверждённому пакету и передаёт его и все последующие еще раз.</li> <li>10. В алгоритме Selective Repeat/Reject в отличие от предыдущего, пакеты не отправляются повторно без необходимости.</li> <li>11. В третьем варианте отправитель передаёт несколько пакетов до конца, в случае ошибки высылает проблемный пакет еще раз.</li> <li>12. После исправления ошибок алгоритм продолжает с того же места, откуда закончил перед обнаружением ошибки</li> </ol>		
<b>Позитивные следствия и/или достоинства описанной в статье технологии (минимум три пункта)</b> <ol style="list-style-type: none"> <li>1. Алгоритмы обнаружения и исправления ошибок просты в реализации.</li> <li>2. Для декодирования используется более простое оборудование (по сравнению с другими технологиями).</li> <li>3. Существует несколько типов алгоритма для систем с более или менее высокой частотой ошибок.</li> </ol>		
<b>Негативные следствия и/или недостатки описанной в статье технологии (минимум три пункта)</b> <ol style="list-style-type: none"> <li>1. При высокой частоте ошибок объём передаваемой информации может оказаться на порядок больше.</li> <li>2. В канале с высокой частотой ошибок сигнал от получателя может быть также потерян, этот случай никак не обрабатывается.</li> <li>3. Нет эффективного алгоритма, который работал бы за минимальное время и отправлял наименьшее количество пакетов.</li> </ol>		