УДК 004.7

В. С. Мельник, А. В. Горячев
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Построение географически-распределенной компьютерной системы и оценка ее надежности

Описан подход к проектированию географически-распределенной системы, позволяющий повысить отказоустойчивость каждого независимого звена сетевой инфраструктуры, минимизировав в целостной системе вероятность потери данных. Целью проведенных исследований является разработка комплексного решения по передаче непрерывного потока данных различными способами, которые, в совокупности, позволяют обеспечить отказоустойчивую передачу потока от вещательных серверов к серверам видеозаписи. Также целью исследований является непосредственный набор скриптов, позволяющих интеллектуально определять ситуацию, при которой каждое независимое звено системы переходит в критическое состояние. При наступлении критического состояния, т. е. состояния, при котором звено системы теряет свою работоспособность, в работу вводится резервное звено. В используемой схеме подключения предлагается повысить отказоустойчивость географически-распределенной системы, при которой каждое из звеньев выступает в качестве независимой единицы компьютерной системы. Проведен анализ различных вариантов подключения системы с учетом вероятности выхода из строя каждого из независимых звеньев.

Отказоустойчивость, надежность, резервирование, шифрование, потоковая передача видео, кодирование видео

В настоящей статье рассмотрено несколько задач, решающих одну большую: передача шифрованного непрерывного потока данных по каналам связи.

1. Обеспечение процесса непрерывного кодирования потока данных различными способами во избежание ошибок, возникающих на этапе кодирования потока: аппаратное и программное кодирование.

2. Процедура резервирования среды передачи информации как на устройстве, передающем поток данных, так и на записывающих узлах инфраструктуры.

3. Обеспечение резервирования записывающего узла за счет применения различных алгоритмов и инструментов приема потока на записывающем сервере с целью обеспечения репликации конечных данных.

С сервера видеопоток передается провайдерам спутниковой связи по технологии SDI, от них – на спутник и далее – к потребителям (кабельным провайдерам и затем – конечному пользователю). По закону Российской Федерации (ст. 34 «О хранении материалов радио- и телепередач») необходимо сохранять запись эфира за последний месяц.

Технология SDI (Serial Digital Interface) – семейство цифровых видеоинтерфейсов для профессионального использования, которые стандартизированы Обществом инженеров кино и телевидения. С помощью этой технологии возможно передавать видеосигнал, а также – до 16 каналов аудио. В настоящее время эта технология используется в сфере телевещания повсеместно [1]. Процесс резервирования источника достаточно прост и стандартен – репликация источника на 2 различных физических устройства.

При построении корпоративных отказоустойчивых систем целесообразно объединение их в единую сеть для возможности передачи стабильного потока данных. Однако в рассматриваемом слу-

чае необходимо обеспечить запись потока данных в нескольких географически-распределенных точках, так как важность имеет как сама система в целом, так и канал, по которому происходит передача данных. Важно отметить, что при необходимости подобная система может работать круглосуточно (даже во время проведения технического обслуживания на каждом из записывающих устройств).

На рис. 1 изображена схема соединения записывающих устройств с источником видеопотока.

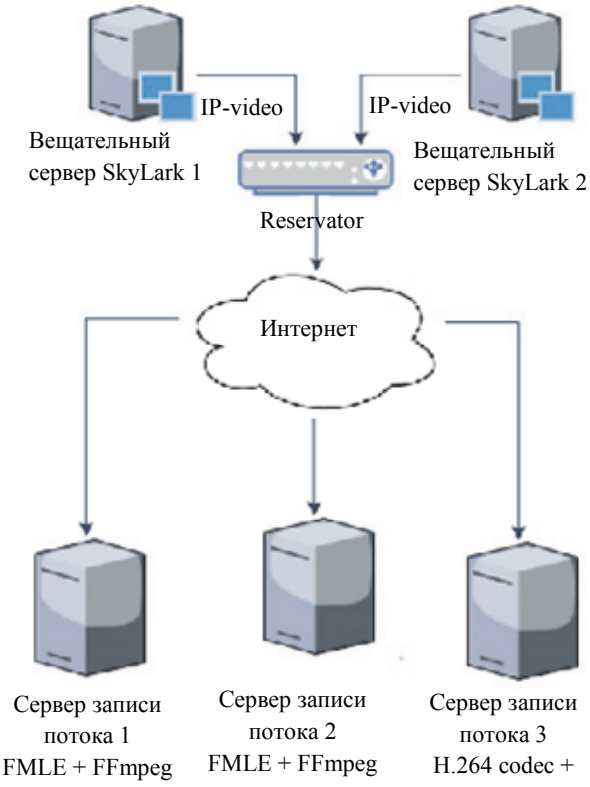

Рис. 1

Сервис, который предоставляет доступ к стратегически важным данным, должен строиться таким образом, чтобы в случае отказа отдельного компонента не произошло нарушения доступа и потери этих данных. На момент написания данной статьи не удалось обнаружить коммерческого либо бесплатного программно-аппаратного решения, которое позволило бы обеспечить отказоустойчивость всех узлов настоящей цепочки. В большинстве случаев при решении описываемой задачи не учитывается как минимум резервирование одного из звеньев всей системы. К тому же – коммерческие решения крайне требовательны к ресурсам и стоимость их велика. Помимо этого необходимо большое количество оборудования и место для его монтажа (спутниковая тарелка, ресивер, подписка на вещание и т. д.).

Рассматриваемое в настоящей статье решение уже используется на производстве одним телеканалом спутникового телевидения.

Стоит отметить, что источник видеосигнала (вещательный сервер) всегда дублируется (это стандарт) и нет ни одного телеканала, который бы производил телевещание без резервного сервера. В схеме на рис. 1 изображены 2 вещательных сервера. Каждый из них в режиме реального времени отдает картинку, которая синхронна с точностью до кадра (вещание происходит по стандарту 25 кадров/с). Это позволяет вводить в схему устройство, которое автоматически переключает сигнал вещания с основного вещательного сервера на резервный в случае отказа первого. К аппаратному анализатору видеопотока (резерватору) подключаются оба вещательных сервера, и их видеопотоки постоянно сравниваются в режиме реального времени. В случае какого-либо несоответствия между потоками устройство автоматически отсылает предупреждение группе технической поддержки. Если один из серверов прекращает свое вещание, происходит автоматическое переключение на вещание со второго – резервного сервера. При возврате работоспособности первому серверу устройство переходит в автоматический режим работы, по умолчанию отправляя видеопоток с сервера, который помечен в системе как основной.

Для решения задачи резервации потока данных, полученного на выходе, используются следующие варианты кодирования:

1. Программный – кодирование потока данных с помощью специального программного обеспечения.

2. Аппаратный – кодирование потока данных с помощью специальной аппаратной разработки (H.264/265 Encoder), позволяющей транслировать аппаратный поток данных, передаваемый через SDI, в поток пакетов, передаваемый посредством Ethernet.

В первом случае кодирование и отправка пакетов происходят с помощью бесплатного программного продукта от Adobe – Flash Media Encoder. Здесь поток данных, принимаемый через SDI, транслируется на устройство-приемник программным способом. Этот вариант реализации хорош тем, что не требует дополнительных финансовых затрат, не требует места в серверных и дополнительного охлаждения. Но этот вариант не лишен главного недостатка – программная реали-

зация передачи имеет свойство накапливать ошибки передачи, и в определенный момент программно-аппаратный комплекс перестает функционировать из-за ошибок, которые накапливаются в процессе работы. Для повышения стабильности работы данного способа разработан скрипт, позволяющий перезапускать программное решение в автоматическом режиме через заданный промежуток времени. Это позволяет избежать накопления большого количества ошибок передачи, приводящих к зависанию передающего сервера. Помимо перезапуска по расписанию скрипт позволяет интеллектуально отслеживать наступление критической ситуации (зависание кодирующего ПО) и перезапускать его в данном случае. Также скрипт позволяет запускать лишь ядро кодирующего модуля, не запуская при этом интерфейс, а значит – потреблять меньшее количество ресурсов.

Во втором случае надежность повышается за счет использования аппаратного кодировщика данных. Кодирование сигнала из SDI в поток данных, передаваемый посредством Ethernet, происходит с помощью аппаратного модуля (Encoder h.265/h.264), который позволяет кодировать поток в режиме реального времени. Данный модуль подключается к передающему серверу посредством разъема SDI, а к сети – через Ethernet (рис. 2).

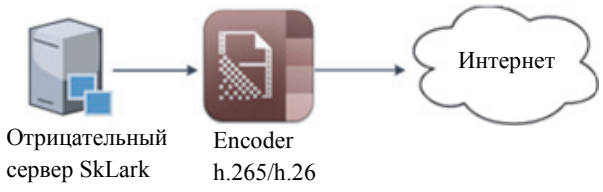

Рис. 2

Настоящий способ также не лишен проблем. При кодировании сигнала таким образом фреймы не могут подаваться напрямую на устройство записи, так как передаются они в кодированном виде, буферизируясь на источнике сигнала, и отправляются группой пакетов через заданный промежуток времени.

В случае решения задачи резервации передатчика сигнала все просто – 2 корпоративных интернет-канала, которые будут работать одновременно: балансировка нагрузки и обеспечение отказоустойчивости. В этом случае, если происходит отключение одного из каналов связи, данные продолжают передаваться далее через оставшийся канал.

В случае резервации приемника все гораздо сложнее: помимо обеспечения отказоустойчивости канала связи необходимо обеспечить и отказоустойчивость приемника. Записывающее устройство географически отделено от передатчика, что позволяет решить сразу две проблемы: резервировать одновременно канал связи и само записывающее устройство посредством создания аналогичного самостоятельного комплекса и установки его в иной географической точке.

Перед решением поставленных задач необходимо «зафиксировать» параметры передаваемого видеопотока для фиксации необходимого дискового пространства для хранения данных на приемниках:

$$\text{Capacity} = \\ = \min\left[f\left(\frac{\text{MaxDpbMbs}}{\text{PicWidthMbs} \times \text{FrameHeightInMbs}} \right) \right],$$

где MaxDpbMbs – константа, которая определяется по таблице максимальных уровней значений свойств; PicWidthMbs – ширина передаваемого кадра; FrameHeightInMbs – высота передаваемого кадра [2].

На сервер передается поток такого же качества, как и на спутник, в соответствии с законодательством Российской Федерации, и спецификация сигнала следующая (приведены необходимые данные для расчета дискового пространства):

– размер изображения: 1920×1080;

– частота кадров: 25 кадров;

– субдискретизация: 4:2:0.

На основе данных можно высчитать примерную емкость необходимого дискового пространства на устройстве для записи потока. Необходимо учитывать, что фактический размер может незначительно меняться как в меньшую, так и в большую сторону. Связано это с процентом изменения наполнения кадра, т. е. насколько сильно (N + 1)-й кадр отличается от N-го кадра. Примерный объем дискового пространства составляет 671.15 Гбайт.

Для географически-распределенной системы необходимо обеспечить безопасную передачу данных. Эта задача решена следующими способами:

1. Для потока данных, который передается программным способом, используется VPN-соединение. В этом случае устройство-передатчик и записывающее устройство соединены виртуальным тоннелем, который позволяет защитить данные от внешних воздействий.

2. Для данных, передаваемых аппаратным способом, используется NAT и ряд правил в Firewall. В этом случае имеется доступ извне к защищенной

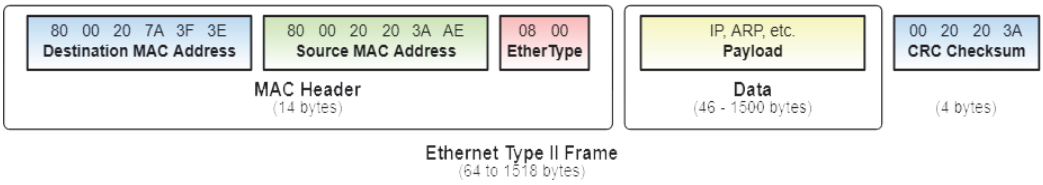

Рис. 3

системе не только с определенных IP-адресов, но и по определенному протоколу. Все остальные доступы извне заблокированы и отброшены.

При использовании VPN-соединения возникает проблема передачи полезной информации. Из-за дополнительного шифрования данных в пакете уменьшается количество данных, которые можно передать за один раз. Это связано с тем, что некоторое количество байт резервируется для создания дополнительного заголовка. Разница в размере пакета отображена на изображении (рис. 3).

В обычном, нешифрованном пакете значение полезной нагрузки варьируется, но максимальный размер – 1500 байт. В случае с шифрованными пакетами это значение уменьшается. В зависимости от типа VPN и шифрования это значение также может меняться:

• PPTP:

$$MTU = MTU_{max} - IP_{head} - IP_{GRE} - IP_{PPP},$$
$$MTU = 1500 - 20 - 4 - 2 = 1474.$$

• L2TP с IPSec-шифрованием:

$$MTU = MTU_{max} - IP_{head} - IP_{UDP} -$$
$$- IP_{L2TP} - IP_{PPP} - IP_{ESP},$$
$$MTU = 1500 - 20 - 8 - 16 - 2 - 20 = 1394.$$

• L2TP с IPSec-шифрованием и NAT-Traversal. В этом случае пакет дополнительно обрабатывается таким образом, чтобы было возможно соединение между различными клиентами через межсетевые экраны:

$$MTU = MTU_{max} - IP_{head} - IP_{UDP} -$$
$$- IP_{L2TP} - IP_{PPP} - IP_{ESP} - IP_{head},$$
$$MTU = 1500 - 20 - 28 - 16 - 2 - 20 - 20 = 1394,$$

где MTU_{max} – максимальная единица передачи; IP_{head} – единица данных, занимаемая заголовком пакета; IP_{GRE} – единица данных, занимаемая заголовком GRE; IP_{PPP} – единица данных, занимаемая заголовком PPP; IP_{UDP} – единица данных, занимаемая заголовком UDP; IP_{L2TP} – единица данных, занимаемая заголовком L2TP; IP_{ESP} – единица данных, занимаемая заголовком ESP.

Исходя из расчетов, можно предположить, насколько увеличивается нагрузка на сеть при передаче данных в круглосуточном режиме. За основу возьмем объем данных, который получает сервер записи за месяц – 671.15 Гбайт данных. Далее – посчитаем количество пакетов, которые потребуются для передачи этого объема данных различными способами (без учета потерь):

1. Без VPN – 469 185 пакетов.
2. PPTP – 477 462 пакета.
3. L2TP с IPSec-шифрованием – 504 866 пакетов.
4. L2TP с IPSec-шифрованием и NAT-Traversal – 515 220 пакетов [3].

Видно, что, при использовании VPN необходимо большее количество пакетов для передачи того же объема данных. Следовательно – нагрузка на канал возрастает приблизительно на 10 %, что негативно сказывается на скорости и качестве передачи.

Помимо задачи передачи данных необходимо решить и проблему приема сигнала. В случае передачи потока данных программным способом на устройстве-приемнике дополнительно нет необходимости установки дополнительных кодирующих программ, так как данные приходят на сервер в первоначальном виде. В случае с аппаратным кодированием все сложнее, так как на принимающей стороне необходимо дополнительно установить и настроить прокси-сервер, который позволит перекодировать данные в корректный формат, а также – послужит буфером, накапливающим в себе данные и одновременно исправляющим ошибки, возникающие при передаче данных.

При аппаратном кодировании записывающее устройство разделено на несколько составляющих:

1. Прокси-сервер. Принимает сырой поток данных от аппаратного источника через незащищенную сеть Интернет. Помимо приемки данных он служит буфером, отвечающим за накопление потока в небольшом количестве, и инстанцией, исправляющей ошибки, которые возникают в процессе передачи данных.

2. Сервер кодирования. Принимает освобожденный от ошибок поток данных и кодирует их в формат, совместимый с записывающим сервером.

3. Сервер записи. Принимает поток данных от сервера кодирования и записывает их на диск.

При программном кодировании данных из этой схемы исключаются первые 2 программных сервера и данные сразу записываются на диск. Недостаток программной схемы заключается в передающем сервере, который не способен безошибочно и безостановочно передавать данные круглосуточно.

Помимо решения проблем, связанных с передачей данных, возникает проблема хранения. Место на записывающих серверах ограничено, и его нужно периодически очищать. Для этого на каждом принимающем сервере стоит программный сервер автоматизированной очистки устаревших данных. По техническому заданию, данные на сервере необходимо хранить 30 дней, после чего они устаревают и более не нужны. Данная задача решается на сервере автоматизации за счет скриптов, написанных на языке программирования bash, и включает в себя сразу несколько проверок:

1. Проверку атрибутов записанных данных на срок давности. Скрипт проверяет дату создания у файлов, полученных с вещательного оборудования. Если дата создания файла превышает указанный в скрипте срок давности, вступает в работу скрипт удаления.

2. Автоматическое удаление. Данный блок команд позволяет удалять устаревшие данные. Скрипт удаляет данные, дата создания которых превышает срок хранения, установленный в системе, в автоматическом режиме.

3. Скрипт проверки изменения размера занимаемого дискового пространства. Данное средство позволяет в автоматическом режиме проверять поступающий поток данных. Если занимаемое дисковое пространство не изменяется (отсутствует входящий поток данных), происходит проверка доступа к Интернету и пересылка сообщения системному администратору.

4. Скрипт проверки работоспособности программных серверов. Постоянно проверяется наличие запущенных процессов в системе. Если какой-либо из процессов отсутствует, происходит автоматический его перезапуск.

Помимо решения основных задач также решена и косвенная – задача быстрой развертки независимого звена сетевой инфраструктуры. Предполагается, что есть вероятность выхода из строя одного из звеньев. При переходе на резервный способ записи необходимо как можно скорее решить проблему основного звена и вернуть ему работоспособность, так как в данном случае резерва уже нет и вся система в целом подвержена краху. Решением в данном вопросе является автоматизированная установка и настройка программного обеспечения на конечном сервере. Инсталляция программных компонент происходит с помощью написанного ранее сценария команд для программного обеспечения Ansible, которое позволяет контролировать ошибки при инсталляции, а также является идемпотентным решением, т. е. при повторном применении операции (команды в данном случае) на конечном сервере позволяет давать такой же результат, что и при первичном выполнении.

Таким образом, для развертывания любого из звеньев достаточно лишь установить операционную систему (Ubuntu Server в настоящем случае) и ansible на одном из серверов. На остальных же – установить демоны ssh и разрешить подключение с управляющего сервера инсталляции пакетов. Далее при выполнении скрипта программа установки сама введет необходимые параметры и установит необходимый для работоспособности набор компонент.

Важно отметить, что помимо «дежурной» проверки работоспособности (с помощью скриптов) проверка выполняется и выборочно с помощью вероятностных алгоритмов, работоспособность которых была проверена ранее. Среди них были выбраны самые действенные и включены в работу [4].

Алгоритм работы прост: на нескольких серверах запущен дополнительный набор скриптов, которые выборочно проверяют доступность серверов записи, а также – их работоспособность. Происходит это с помощью генетического алгоритма.

Генетический алгоритм основывается на механизмах, аналогичных естественному природному отбору.

Любой организм в данном случае представлен своим уникальным фенотипом, который фактически определяет, чем является объект в настоящем мире, и генотипом, который содержит всю необходимую информацию об объекте на уровне набора хромосом. При этом каждый ген (элемент) информации генотипа имеет свое отражение в

фенотипе. Таким образом, для решения подобных задач необходимо представить каждый признак объекта в форме, подходящей для использования в генетическом алгоритме.

В настоящем алгоритме помимо обычного математического аппарата также присутствует и вероятностная составляющая, которая делает алгоритм гибким, удобным и способным подстраиваться под конкретную ситуацию. Схема алгоритма проста:

На начальном этапе происходит инициализация: случайным образом формируется начальная популяция, параметры которой полностью настраиваются пользователем. После создания начальной популяции вычисляется приспособленность каждой особи и популяции в целом по формулам:

$$F_{A_i} = \text{fit}(A_i),\ i = 1, ..., k;$$
$$F_t = \text{fit}(B_i),\ i = 1, ..., k,$$

где A_i – особь из популяции A; B_i – особь из популяции B; k – количество особей в популяции.

Если популяция окажется неудачной в целом, она заменяется другой. Дальнейшая работа алгоритма строится на основе проверок особей и приближении к решению поставленной задачи. Для этого производится сравнение особей между собой, скрещивание особей и их мутация. В конечном итоге получается набор новых особей и новая популяция, которая превосходит старую за счет лучшей приспособленности [5].

Алгоритм вероятностным образом проверяет выборочные звенья цепи. Это позволяет уменьшить вероятность краха системы. С помощью написанного сценария для ansible производится сбор информации с выбранного сервера – информации о наличии свободного места на жестких дисках, информации о температуре процессора и ряда других важных параметров. Если система проверки выявляет отклонения, группе инженеров отправляется сообщение об отклонении хотя бы одного параметра от заданных значений. В этом случае важную роль играет идемпотентность, которая предотвращает повторное внесение каких-либо изменений в систему, если код исполнялся ранее. Другими словами, при попытке повторно изменить ряд параметров, файлов или еще каких-либо значений на аналогичные программа выполнения не выдаст ошибку, а отобразит в логах информацию о том, что запрашиваемое изменение уже применено. Это свойство не относится лишь к сбору информации: любые свойства системы отображаются при каждом выполнении скрипта [6].

В статье описан подход, позволяющий принимать непрерывный поток данных от вещательных серверов к серверам записи. Настоящий способ коммутации уникален тем, что система в автоматическом режиме распознает наступление критической ситуации, а также – предугадывает ее наступление. Основываясь на вышеизложенном, стоит отметить, что лучше всего в системе использовать способ аппаратного кодирования, обеспечивающий большую безопасность при передаче данных. Дальнейшей перспективой развития предложенной схемы подключения является внедрение в систему интеллектуальных алгоритмов, описанных в [4]. Считаем, что результаты, предложенные в данной статье, могут быть востребованы разработчиками программного обеспечения, занимающимися созданием программ для записи потокового видео.

СПИСОК ЛИТЕРАТУРЫ

1. Charles A. Poynton. Digital Video and HDTV. Waltham: Morgan Kaufmann, 2003. P. 694.

2. Gary J. Sullivan, Pankaj Topiwala and Ajay Luthra. The H.264/AVC Advanced Video Coding Standard. URL: 2004http://www.fastvdo.com/spie04/spie04-h264OverviewPaper.pdf (дата доступа 23.01.19).

3. Detailed VPN Comparison Chart. 2019. URL: https://thatoneprivacysite.net/vpn-comparison-chart (дата доступа 16.12.18).

4. Мельник В. С. Исследование алгоритмов для решения задачи маршрутизации пакетов в компьютерной сети // XXI Междунар. конф. по мягким вычислениям и измерениям (SCM-2018). СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2018. С. 673–676.

5. Poli R., Langdon W. B., McPhee N. F. A Field Guide to Genetic Programming. 2008. URL: http://www0.cs.ucl.ac.uk/staff/W.Langdon/ftp/papers/poli08_fieldguide.pdf (дата доступа 11.11.18).

6. Gunawardena Jeremy. An introduction to idempotency. Cambridge: Cambridge University Press, 1994, 456 с.

V. S. Melnik, A. V. Goryachev
Saint Petersburg Electrotechnical University «LETI»

BUILDING OF A GEOGRAPHICALLY DISTRIBUTED COMPUTER SYSTEM AND ASSESSMENT OF ITS RELIABILITY

This article describes the approach to the design of a geographically distributed system, which allows you to increase the resiliency of each independent network infrastructure, minimizing the probability of data loss in a complete system.
The purpose of the research is to develop algorithms and mechanisms that allow replication of each node of the network infrastructure, as well as intelligently determine the situation in which each independent link in the system goes into a critical state. When a critical state occurs, that is, a state in which the system link loses its operation, a backup link is put into operation.
In the model used, it is proposed to increase the resiliency of a geographically distributed system whereby each of the links acts as an independent unit of a computer system. As a result of the work, an analysis was made of various options for connecting the system, considering the probability of failure of each of the independent links.

Resiliency, reliability, reservation, encryption, video streaming, video encoding

УДК 681.54

О. И. Брикова, С. Е. Душин
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Исследование влияния температуры среды на биологические процессы в моделях типа ASM1

Существующие модели, описывающие процесс биологической очистки, не учитывают влияние внешних факторов, поэтому представляет интерес исследование влияния температуры среды на процессы нитри- и денитрификации. В статье представлены нелинейные математические модели нитрификации и денитрификации с учетом влияния температуры среды. В основу данных моделей положены широкоизвестные модели типа ASM1. Зависимость скорости роста микроорганизмов от температуры описывается уравнением Вант-Гоффа. На основе разработанных математических моделей были построены компьютерные модели в среде MATLAB/Simulink. Решаемые задачи заключаются в анализе влияния температуры внешней среды на поведение моделей нитрификации и денитрификации. Приведены семейства графиков при различных температурных режимах. В результате исследований найдены наилучшие температурные диапазоны для развития биоценоза активного ила, которые могут быть положены в основу создания математической модели типа ASM с учетом температурного фактора.

Биологическая очистка, активный ил, нитрфикация, денитрифкация, математическое моделирования, температура внешней среды

Развитие городов, промышленности, сельского хозяйства неизменно приводит к необходимости усовершенствования систем очистных сооружений. Метод биологической очистки микроорганизмами активного ила – это один из распространенных и эффективных методов очистки сточных вод [1], [2]. Биоценоз активного ила потребляет субстрат и преобразует его в безопасное для окружающей среды состояние.

Разработка и исследование адекватных динамических моделей управляемых биохимических процессов – это один из основных этапом проектирования систем управления технологическими процессами очистки сточных вод. Такая необходимость в математическом моделировании в первую очередь связана с жесткими требованиями к очищенной воде, сложностью при проведении экспериментов на реальном объекте, небольшим набором средств измерения и продолжительными лабораторными исследованиями, а также сложностью качественного анализа биоценоза активного ила [3]. На сегодняшний день ма-