

СРЕДСТВА ПОВЫШЕНИЯ НАДЕЖНОСТИ И ДИАГНОСТИКИ ЭЛЕМЕНТОВ КОМПЬЮТЕРНЫХ ПРОМЫШЛЕННЫХ СЕТЕЙ

Мальчева Р.В., Арутюнян А.Р., Завадская Т.В.

Кафедра ЭВМ, ДонНТУ

Email: raisa@cs.dgtu.donetsk.ua, arutyunyan@cs.dgtu.donetsk.ua

Abstract

Malcheva R.V., Arutyunyan A.R., Zavadskaya T.V. Elements for the increase stability and for the testing of components of industrial computer networks. In the paper the mathematical model of industrial computer network and the algorithm to test the components of industrial computer networks are described.

Общая постановка проблемы. Процесс автоматизации технологических процессов производства развивается все более ускоряющимися темпами: увеличивается количество «интеллектуальных» конечных устройств, растет число вовлеченных в процессы контроля и управления технологическим процессом вычислительных систем на базе микроконтроллеров. В этих условиях существенно возрастает роль данных, собираемых на всех уровнях АСУ ТП [1 - 5].

В данный момент актуальной задачей является автоматизация малых и средних по величине заводов, в т.ч. построение системы диспетчерского контроля. Такие системы обычно включают ряд узлов промышленной автоматики, которые ведут непрерывную регистрацию технологического параметра [4, 5]. В связи с большой территориальной распределенностью таких систем очень остро встает вопрос поиска неисправности элементов системы. Кроме того, в ряде случаев, элементы сбора информации находятся в трудно доступных или опасных для жизни местах. Поэтому одной из актуальных задач является разработка алгоритмов и способов удаленного тестирования элементов сети, а также алгоритмов построения сетей с возможностью тестирования [4 - 6].

Зарубежные фирмы, такие как SIEMENS, ACP-DAC, ADVANTECH, предлагают инструментарию, позволяющие решить данную задачу [1, 2], но их применение предполагает использование стандартов и оборудования соответствующих фирм [6 - 8]. Выполнение этих требований не всегда приемлемо по ряду причин: закрытые стандарты, трудное объединение элементов разных фирм производителей, дороговизна оборудования, удаленная поддержка [3 - 6]. Открытые стандарты, например, MOD-BUS, поддерживаемые большинством отечественных производителей, не предполагают тестирования элементов сети. Следовательно, разработка алгоритмов тестирования с последующей интеграцией их в передовые открытые сетевые стандарты является актуальной задачей.

Математическая модель компьютерной промышленной сети

Промышленную компьютерную сеть будем представлять в виде графа со взвешенными ребрами и вершинами $G = (X, A)$. Ребрам $a_i \in A$ приписаны весовые вектора C_i , а вершинам $x_i \in X$ приписаны веса d_i .

Определение. Цепь графа G называется *рассогласованной*, если для данной цепи выполнимо условие $\frac{B}{A} > K$, где K - порог согласования линии.

Определим базовые весовые вектора ребер:

$$C_i = \{l, \gamma, Z_B\}, \quad (1)$$

где l – длина сегмента сети соответствующего ребру;
 γ – коэффициент затухания или распространения;
 Z_B – волновое сопротивление кабеля.

Определим базовые весовые вектора вершин

$$P_i = \{d, \pi, p, s, n, Z_H\}, \quad (2)$$

где d – тип устройства;
 π – стационарное распределение;
 p – количество портов;
 s – скорость обмена данными (количество пакетов в ед. времени);
 n – тип сетевого интерфейса;
 Z_H – сопротивление нагрузки.

Для типа устройства d_i определим диапазон принимаемых значений [1..6],

где $d_i = 1$ соответствует промышленному маршрутизатору;
 $d_i = 2$ соответствует пассивному расширителю;
 $d_i = 3$ соответствует промышленному Ethernet контроллеру;
 $d_i = 4$ соответствует узлу сети;
 $d_i = 5$ соответствует интеллектуальному преобразователю интерфейсов;
 $d_i = 6$ соответствует порту узла сети.

Узлы, для которых $d_i \neq \{6\}$, представлены в виде k -ичных деревьев глубиной 1, где k – количество портов узла.

Приведем на рис.1а) экспериментальную подсеть, внедренную на предприятие ООО «Укрвторресурсы». Представим на рис.1б) математическую модель экспериментальной сети, описанную в виде графа.

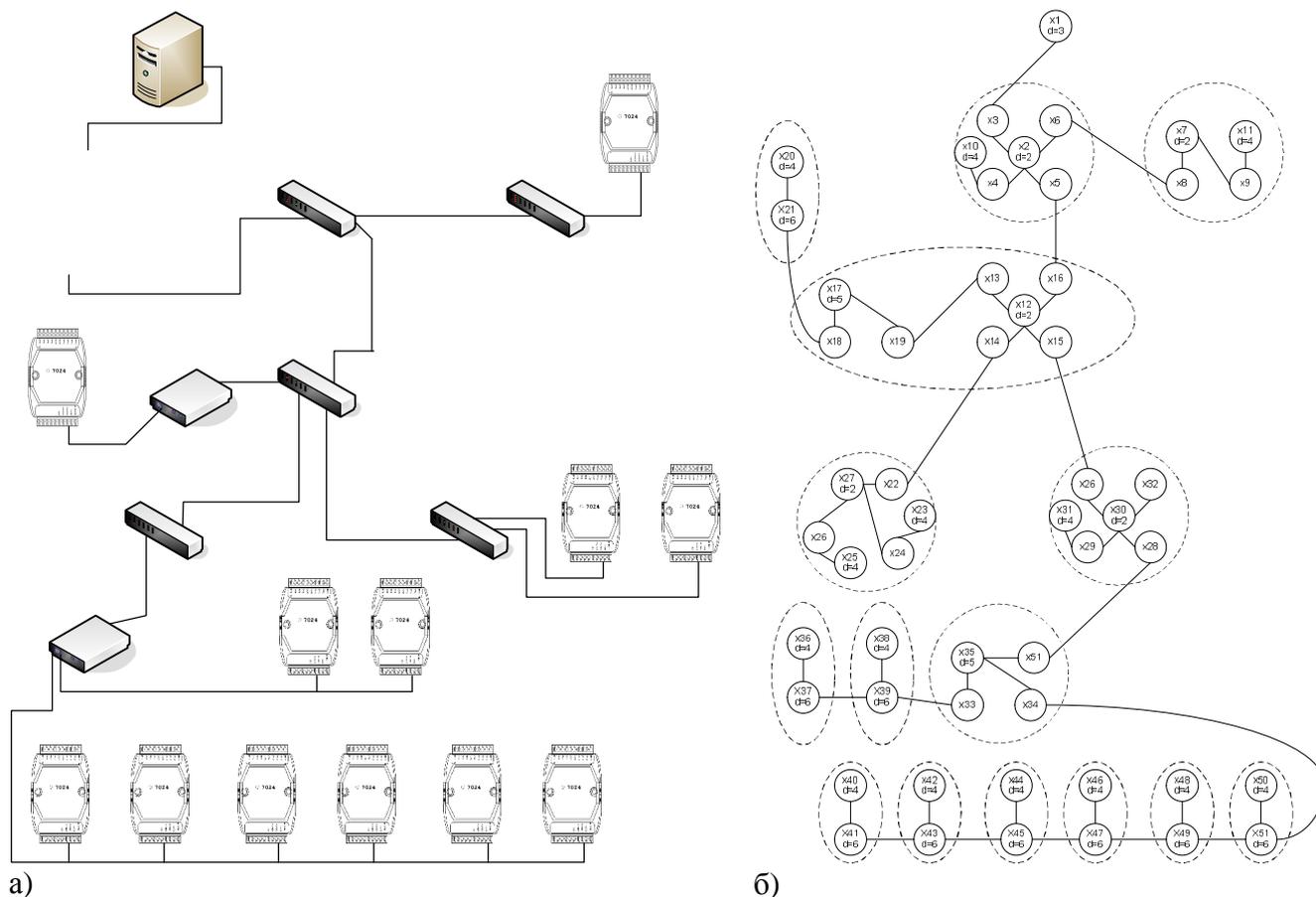


Рис. 1. Экспериментальная подсеть:
 а) топология экспериментальной подсети;
 б) математическая модель экспериментальной подсети.

Перед описанием алгоритма приведем несколько определений.

Пусть вершина x_1 является вершиной, с которой начинается опрос всей ЛВССД, следовательно вершина x_1 является корнем дерева сети.

Определение 1. Ребро $a_i \in A$ графа $G = (X, A)$ существует, если вероятность потери данных через сегмент сети, соответствующий ребру, меньше P_{oui} .

Определение 2. Если возможно получить данные от узла сети, соответствующего вершине x_i , то будем считать, что существует путь из вершины x_1 к вершине x_i .

Определение 3. *Локальный подграф* – подграф, внутри которого ребра (сегменты сети) не могут быть удалены (повреждены).

Примеры локальных подграфов приведены на рис.1б) и обведены пунктирными линиями.

На рис.1б) ребра, которые не могут быть разрушены, соответствующие сегментам сети, которые не могут быть повреждены, - это ребра (x_4, x_{10}) , (x_{23}, x_{24}) и др.

К локальным подграфам можно отнести подсеть, установленную в железный щит, в котором сегменты проводов, соединяющие узлы сети, не могут быть повреждены извне.

Определение 4. *Внешней вершиной* локального подграфа назовем вершину, инцидентную ребру, не входящему в локальный подграф.

Определение 3 не относится к ребрам, которые инцидентны внешним вершинам.

Определение 5. Граф с древовидной структурой описывает сеть с возможностью самотестирования, если вершины, для которых $d_i=4$, являются только листьями дерева. Назовем такой граф – *граф тестирования*.

Определение 6. *Граф автоматического тестирования* – граф тестирования, у которого вершины с весами $d_i=6$ инцидентны не более чем двум вершинам.

Лемма 1. В *графе тестирования* не существует пути из вершины x_i в любую вершину $x_j \in X$, для которой $d_i \neq 6$, без захода в вершину $x_j \in X$, для которой $d_j=6$ и которая инцидентна вершине $x_k \in X$, $d_k \neq 6$, $d_k \neq 4$.

Алгоритм автоматического тестирования

Алгоритм диагностики представляет собой алгоритм локализации выпавшего ребра или ребер графа. Т.к. мультипортовые узлы представлены в виде подграфов, алгоритм позволяет определить вышедшие из строя порты узлов. Диагностика ребер происходит в процессе опроса узлов графа, при обнаружении выпавшего ребра активизируется запасное ребро и граф остается связанным.

Шаг 1. Определим множество вершин Ψ , к которым не существует путей из вершины x_1 , для которой $d_i=4$.

Например, для графа изображенного на рис.1б, $\Psi = \{x_{11}, x_{23}, x_{25}, x_{20}, x_{38}, x_{36}, x_{46}\}$.

Шаг 2. Составим множество ребер Ω , по которым не осуществлялся обход.

Для нашего примера $\Omega = \{x_2 - x_6 - x_8 - x_7 - x_9 - x_{11}, x_{12} - x_{14} - x_{22} - x_{27} - x_{24} - x_{23}, x_{27} - x_{26} - x_{25}, x_{17} - x_{18} - x_{21} - x_{20}, x_{35} - x_{33} - x_{39} - x_{38}, x_{39} - x_{37} - x_{36}, x_{47} - x_{46}\}$.

Шаг 3. Удалить из Ω неразрушимые ребра локальных подграфов.

В нашем случае $\Omega = \{x_2 - x_6 - x_8 - x_7, x_{12} - x_{14} - x_{22} - x_{27}, x_{17} - x_{18} - x_{21} - x_{20}, x_{35} - x_{33} - x_{39} - x_{38}, x_{39} - x_{37} - x_{36}, x_{47} - x_{46}\}$.

Шаг 4. Если $\Omega = \emptyset$, то шаг 10, иначе шаг 5.

Шаг 5. Определим самую короткую цепь, состоящую из ребер множества Ω .

Для нашего примера это $x_{47} - x_{46}$.

Шаг 6. Если цепь состоит из одного ребра, то выдать локализацию неисправности, характер неисправности – отсутствие питания узла или неисправность порта.

Шаг 7. Если цепь состоит из 3-х ребер, то с помощью ближайшего узла, для которого $d_i=1$, произвести тестирование в вершине, соответствующей порту. Согласно определению 5 и лемме 1 такая вершина всегда существует. Тестирование в узле, соответствующем порту, позволяет определить длину сегмента, присоединенного к порту, и, следовательно, целостность ребра, соответствующего линии передачи данных. Также такое тестирование определяет работоспособность порта. Тестирование порта осуществляется посылкой данных в режиме эха. После такого тестирования легко локализовать неисправность сети. Подробный алгоритм тестирования приведен ниже.

Шаг 8. Удалить из множества Ω ребра, входящие в пути (цепи), проходящие через локализованное неисправное ребро.

После шага 8

$\Omega = \{x_2 - x_6 - x_8 - x_7, x_{12} - x_{14} - x_{22} - x_{27}, x_{17} - x_{18} - x_{21} - x_{20}, x_{35} - x_{33} - x_{39} - x_{38}, x_{39} - x_{37} - x_{36}\}$

Шаг 9. Перейти к шагу 4.

Шаг 10. Конец.

Средства повышения надежности промышленной компьютерной сети

Схема интеллектуального преобразователя интерфейсов (ИПИ) представляет собой модернизированную схему преобразователя интерфейсов с формированием фрейма. Отличие предложенной аппаратной реализации от известной состоит в контроле формирования как входного, так и выходного фрейма, реализацией системы приоритетного разрешения конфликтов на шине, определении расстояния до следующих узлов подключенных к преобразователю.

Функциональная схема приведена на рис. 2.



Рис. 2. Функциональная схема интеллектуального преобразователя интерфейсов.

Опишем принцип работы интеллектуального преобразователя интерфейсов (ИПИ). При поступлении последовательности бит передаваемого по сети байта на вход ИПИ, она принимается во внутренний буфер процессора. Процессор по принятой последовательности бит определяет скорость передачи данных и формирует буфер для передачи. При поступлении байта в буфер на передачу он сразу начинает передаваться. При параллельной работе приема и передачи информации запаздывание передаваемой информации будет равно времени передачи одного байта на всю серию передаваемых байт.

Анализ наличия устройств на шине и определение длины кабельной линии подключенной к порту ИПИ осуществляется следующим образом. ИПИ переводит шину в нулевое состояние, т.е. на шине устанавливается потенциал равный нулю, инициализирует внутренний счетчик времени и прерывание для определения момента принятия ответного сигнала. Далее ИПИ посылает короткий импульс длиной менее 1 мкс, и ожидает

срабатывания прерывания от принятого отраженного сигнала. После принятия отраженного сигнала анализируется время прохождения сигнала и соответственно определяется длина кабельной линии.

Достоинство описанной схемы и алгоритма работы заключается в том, что внедрение данной возможности в существующие решения не требует дополнительной переделки схемы, а легко адаптируется к реализованной схеме посредством изменения микропрограммного кода. Недостатком является малая мощность выходного импульса, которая не позволяет определять большое количество устройств. Так же при достаточно малом быстродействии процессора менее 8 МГц, становится трудоемким определение более одного устройства.

Доработанная структура ИПИ, которая учитывает описанные выше недостатки, приведена на рис. 3.

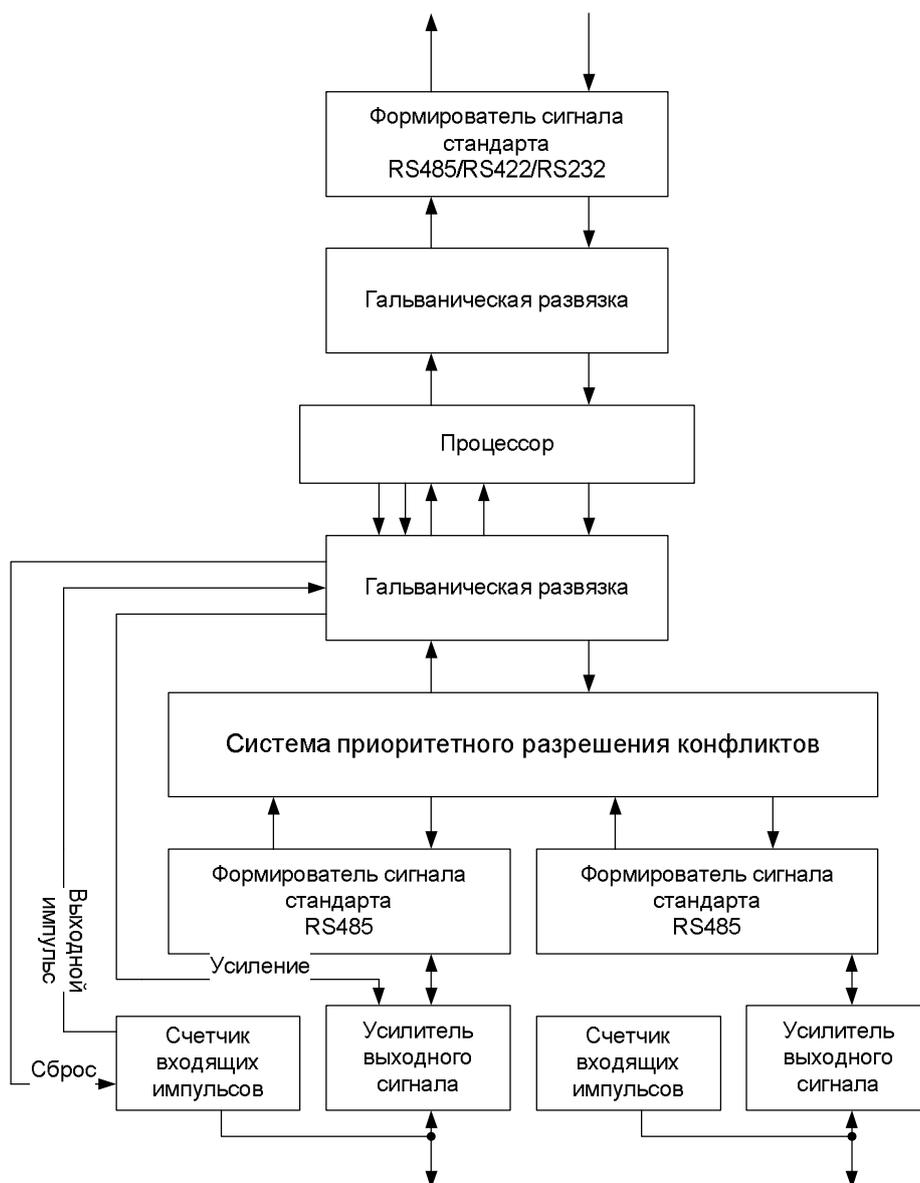


Рис. 3. Функциональная схема интеллектуального преобразователя интерфейсов с выходными сигналами повышенной мощности.

Отличие функциональной схемы на представленной рис. 3 от функциональной схемы на рис. 2 состоит в наличии счетчика входящих импульсов и усилителя входного сигнала.

Анализ наличия устройств на шине и определение длины кабельной линии подключенной к порту ИПИ такого типа осуществляется следующим образом. ИПИ активирует схему усиления и разрешает усиление сигнала на шине, далее переводит шину в нулевое состояние, инициализирует внутренний счетчик времени и прерывание для определения момента принятия ответного сигнала, сбрасывает счетчик входных сигналов длительным присутствием низкого уровня на входе сброса. Далее ИПИ посылает короткий импульс длиной менее 1 мкс, и ожидает срабатывания прерывания от счетчика импульсов. Счетчик импульсов считает количество принятых отраженных импульсов, после сброса счетчика длительным импульсом, счетчик считает количество импульсов равное одному, далее формирует сигнал принятия первого импульса и пропускает импульсы приходящие позже. После принятия отраженного сигнала системой прерываний процессора, процессор анализирует время прохождения сигнала и соответственно определяется длина кабельной линии. Первая посылка позволяет определить расстояние до первого устройства на шине. Перед второй посылкой кратковременного импульса сканирования, коротким импульсом сбрасывается состояние счетчика и увеличивает количество принимаемых импульсов на один, т.е. счетчик сформирует выходной сигнал после того, как на его вход поступят два импульса. Т.о. определяется расстояние до второго прибора и т.д.

Второй вариант реализации ИПИ аппаратно сложнее чем первый, но позволяет реализовать функцию сканирования большего количества устройств на более длинных линиях передачи данных.

Выводы

В статье описаны элементы и принципы построения математической модели топологии промышленной компьютерной сети. Предложен алгоритм позволяющий определить сбойный элемент сети. Использование алгоритма позволит точно локализовать не только сбойный узел, а и причину сбоя, такую как, обрыв кабеля или выход из строя порта. Предложена аппаратная реализация элементов тестирования промышленной компьютерной сети, применение которых позволяет повысить надежность сети, локализовать неисправность и использовать резервные каналы для восстановления работоспособности сети.

Литература

1. Кругляк К.С. Промышленные сети: цели и средства // СТА. – 2002. - №4. –С. 6-17.
2. Гусев С. Краткий экскурс в историю промышленных сетей. // Компоненты и технологии. - 2000. - №10. - С. 45-49.
3. Третьяков С.А. Локальные сети микроконтроллеров. // Электроника. - 1999. - №9. – С. 56-61.
4. Густав Олссон, Джангуидо Пиани. Цифровые системы автоматизации и управления. – СПб.:Невский Диалект, 2001. – 557 с.:ил.
5. Николайчук О.И. Системы малой автоматизации. – М.: САЛОН-Пресс, 2003. – 256 с.
6. Сахнюк А.А., Литвин А.М. Промышленные сети. // ПиКАД. – 2004. - №2. –С. 6-8.
7. Гупта А., Каро Р. FOUNDATION FIELDBUS или PROFIBUS_PA: выбор промышленной сети для автоматизации технологических процессов // СТА. - 1999. - №3. –С. 16-20.
8. Кругляк К.С. Практика построения промышленных сетей на базе AS-интерфейса // СТА. -2002. - №4. – С. 30-39.