Improved Quality of Protocol Testing Through Techniques of Experimental Design

Kirk Burroughs, Aridaman Jain, and Robert L. Erickson

Bell Communications Research
331 Newman Springs Road, Red Bank, NJ 07701 U.S.A.
+1 908 758-5951

Abstract

This paper describes how both the quality and efficiency of protocol testing were improved by using a new Bellcore tool called the Automatic Efficient Test Generator (AETG). The AETG tool is based on ideas from experimental design and it creates a test set that contains all possible pairs of involved factors. Two examples are given to illustrate this technique and compare it with traditional approaches. The improved quality of testing leads to a faster detection of non-conformances and a higher quality of products in a shorter development interval. Although the application discussed in this paper covers protocol conformance testing, the techniques for improving the quality of testing can be applied to other types of testing such as feature testing and interoperability testing between two different network elements.

1. Introduction

In protocol conformance testing, a particular switch under test[1] is exercised to verify conformance to requirements contained in a particular Bellcore Technical Reference (TR). However, resources are not always available to carry out a comprehensive conformance testing to the requirements. This paper describes how both the quality and the efficiency of protocol testing have been improved by using experimental design techniques that provide the maximum possible information from as few tests as possible. These techniques were used in the development of a test plan for "Primary Rate ISDN (PRI) Call Control Requirements" (see TR-NWT-001268 [1]). The benefits obtained by applying these techniques to the test plan will be directly translated into the test script development process and finally the actual testing. Although the application discussed in this paper covers protocol conformance testing, the techniques for improving the quality of testing can be applied to other types of testing such as feature testing and interoperability testing between two different network elements.

Section 2 describes the experimental design technique used and why we chose it along with an overview of the testing process. Section 3 discusses the traditional method of testing along with how the experimental design technique can be used to increase the quality of test coverage and decrease the time spent on testing. Section 4 gives two examples to illustrate the application of the experimental design technique to protocol conformance testing and how this technique compares with traditional approaches. Section 5 discusses the extent of reduction in the total number of test cases achieved by the experimental design approach along with the value added by applying this technique.

2. Experimental Design Techniques

Typically, protocol conformance testing consists of a collection of tests that are defined as a combination of various parameters of interest (called factors) where each parameter can take several values (called levels). Let K be the number of factors and L_i be the number of levels for factor i (i = 1,2,...,K). Then the total number of possible tests (i.e., the total number of combinations of K factors) is $N = L_1 \times L_2 \ldots \times L_K$. For moderate or large values of K and/or L_i's, the total number of possible tests N will be quite large and generally, resources are not available to perform all N tests. Therefore, it is necessary to choose a manageable subset containing n tests for inclusion in the test plan, where n is much smaller than N.

Techniques of experimental design address the problem of choosing an efficient subset of test cases. One such technique that has been applied very successfully in a variety of industries is Orthogonal Arrays (OA). OA provides a design containing a minimum number of tests for achieving a complete balance among various combinations of factors (see Taguchi [2], Phadke[3]). Generally, the OA technique is easy to apply when L_i's are equal. Since in our applications (and most other applications encountered in practice), the L_i's are far from being equal, it would have been necessary to put in a significant amount of effort in redefining the test factors and levels to come up with a structure suitable for the application of the OA technique.

---

[1]. Protocol conformance testing can be performed on any system that claims conformance to a particular protocol. This paper focuses on conformance testing for end office switches claiming Primary Rate ISDN conformance.

0-7803-1825-0/94 $4.00 © 1994 IEEE

Even after such a redefinition, an OA design may not exist.

Because of the difficulty of constructing OA in our applications of far from equal L_i's, we use the Automatic Efficient Test Generator (AETG) prepared and developed by Siddhartha Dalal and Gardner Patton of Bellcore. The AETG is based on a new Bellcore algorithm which uses ideas from statistical experimental design and creates a test set that contains all possible pairs of factors. If the L_i's are equal, then the test design produced by the AETG tool could be identical to an OA design. The AETG tool generates a test design that ensures the inclusion of all pairwise combinations, but it does not require a complete balance among various combinations of factors. The sacrifice of complete balance is a small price to pay for obtaining a tremendous reduction in the number of test cases as compared to the OA approach that requires a complete balance. The AETG tool has been used to obtain an initial test design for the two applications discussed in Section 4.

An important property of the initial test design obtained from the AETG tool is that all pairs of factors (L_i x L_j) are represented at least once. However, in cases of factors with a smaller number of levels, the initial test design had several cases of major imbalances in terms of the number of repetitions of various levels. We enhanced the initial design by reducing the number of occurrences of a level with higher than average frequency and increasing the number of occurrences of another level with lower than average frequency; these changes improved the balance across different levels of the factor involved. This process of enhancement was tedious and time consuming since we had to examine the test design in several dimensions and ensure that there was no deterioration in the test design in any other respect as a result of these changes. In the future, algorithms will be developed to automate the enhancement process.

3. Protocol Conformance Testing

In protocol conformance testing, a particular switch under test is exercised to verify conformance to requirements in a particular TR. The current process involves writing the test plan, developing the test scripts (often software for automated test systems), executing the tests scripts, analyzing the data and finally reporting the results. This task can be quite complex and time consuming. A factor contributing to the complexity of this task is the number of tests needed to provide sufficient coverage. Traditionally, "sufficient coverage" was determined by the judgment of the test plan author. Available time and the scope of the TR are also major factors in this process.

At Bellcore, several automated test tools currently exist for use in protocol conformance testing. The test sets generated by the AETG tool are easily implemented into the software of these test tools. Perkinson [4] has discussed an earlier attempt at incorporating experimental design techniques for selecting a subset of test cases for protocol conformance testing using an automated test tool. This paper describes the actual use of such techniques for several cases of protocol conformance testing. A configuration of a typical automated test tool is shown in Figure 1.

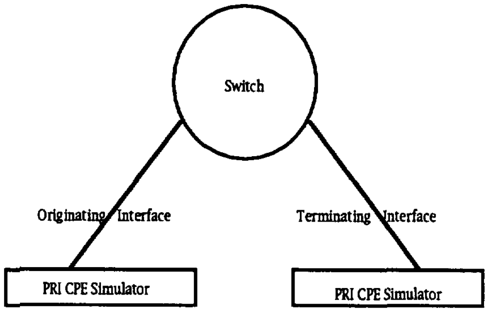

Figure 1. Typical configuration for protocol conformance testing

In a test plan, the tests are typically subdivided into small groups based on some logical or functional separation. In this paper, these groups will be referred to as families. These families are described by a set of parameters, organized into factors and levels.

When discussing the tests (either for a family or for the test plan), two important factors are the number of tests involved and the breadth of coverage (i.e., how evenly is the coverage spread over the parameters involved). Traditional testing approaches often do not provide sufficient breadth of coverage. By applying the AETG tool to the testing process one can improve the breadth of coverage along with reducing the size of the test set in a systematic manner.

Three possible traditional approaches are used as reference points for the AETG tool. First, is the comprehensive approach that includes all test cases. Second, if the comprehensive approach produces a test set that has an unmanageable size, a lower criterion is set: Typically, inclusion of the pairwise combinations of the two most significant factors. Third, if the test set size for the second approach is still unmanageable, an even lower criterion is set: Typically, inclusion of all levels of each factor while the remaining factors are held fixed. The first approach is viewed as being too expensive and the third approach is viewed as having a low quality of coverage.

In the second traditional approach, the test design would include: (i) all pairs for the two most significant factors by fixing the levels of the other factors and (ii) a few more tests to cover the non-fixed levels of the other factors by fixing the levels of the two most important factors. An example of the second traditional approach for the call rejection application is given in the appendix. The test design obtained by applying the AETG tool to our two applications offers the following three significant advantages over the second traditional approach used in the past without increasing the size of the test set:

(i) Wider breadth of coverage of the test factor combinations without leaving any systematic hole in the coverage of the complete space (the second traditional approach has significant holes in the coverage).

(ii) Coverage of all pairs of factors (not just one set of pairs of the two most significant factors as in the second traditional approach)

(iii) Approximate balancing of the frequency of occurrences of various combinations of levels for each pair of factors (no balance in the second traditional approach due to fixing of levels of several factors).

4. Examples

This section discusses two examples that illustrate the application of the AETG tool to protocol conformance testing: Call Rejection and Channel Negotiation. Table 1 summarizes the number of test cases needed and the breadth of coverage for the first two traditional approaches and the AETG approach. Of course, the first traditional approach includes a comprehensive coverage of the test space, but it requires a large number of test cases. The second traditional approach achieves a significant reduction in the number of test cases, but it suffers from a lower breadth of coverage. The AETG approach provides a much broader coverage of the test space with a number of test cases smaller than that for the second traditional approach.

4.1 Call Rejection

Call rejection is defined [1] by the called user responding to the incoming ISDN SETUP message with a RELease COMPlete message. However, any call clearing message can be used to reject a call. The call rejection message is expected to contain a cause value indicating why the call was rejected. The called user can also reject the call before or after sending a CALL PROCeeding message to the Stored Program Control Switching System (SPCS), or before or after timer T303 expires. All these factors plus the bearer capability requested determine the treatment applied to the users involved. Figure 2 shows a call flow diagram for call rejection.

Thus, the factors and levels determining treatment for call rejection are:

- Cause Value (7),
- Bearer Capability (6),
- Call Clearing Message (3),
- Call Rejection before or after the called party transmits a CALL PROCeeding message (2),
- Call Rejection before or after timer T303 has expired (2).

Table 1

Number of Test Cases and Breadth of Coverage

Family	Number of Tests Cases Needed			Breadth of Coverage(%)[2]		
	Trad. #1	Trad. #2	AETG	Trad. #1	Trad. #2	AETG
Call Rejection	504	46	42	100	33	100
Channel Negotiation	108	21	18	100	30	100

---

[2] The Breadth of coverage is defined here as the percentage of all pairwise combinations of the test factors in the family. Traditional approach #1 and the AETG approach include all pairwise combinations of the test factors while traditional approach #2 includes only a fraction of all pairwise combinations. While traditional approach #1 includes all possible test cases, the AETG approach includes a small fraction of all test cases.

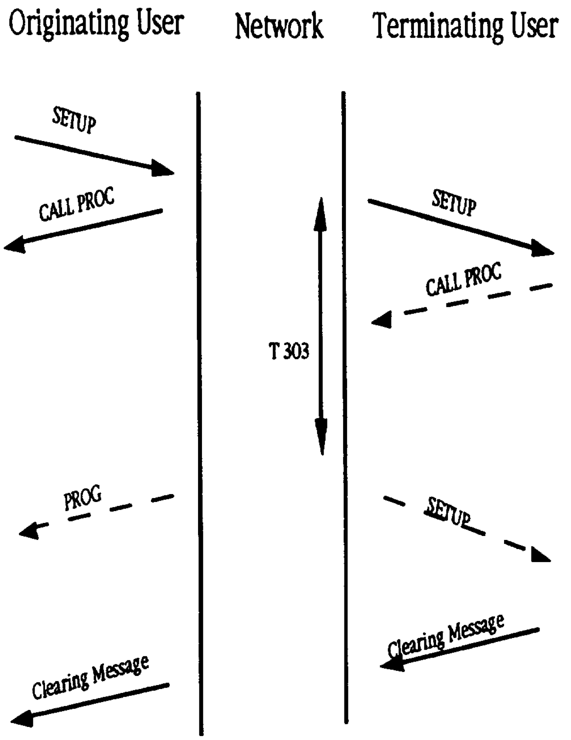

Figure 2. Call Flow Diagram for Call Rejection

To test all combinations (i.e. the first traditional approach) of the above factors requires 504 (7*6*3*2*2) tests. The initial test design obtained by applying the AETG tool provided coverage of all pairwise combinations in 42 tests. However, there were major imbalances with respect to some factors. Since we desired as much balance as possible, the initial test design was changed with respect to these factors to improve their balance. The improved design shown in Table 2 contains 8% of all possible tests.

By comparison, the second traditional approach would produce a set of 46 tests: (i) Forty two tests as pairs of cause and bearer capability while holding the other 3 factors fixed and (ii) 2 tests for the other two possible values for call clearing messages, one test for the other possible value for CALL PROCeeding, and one more test for the other possible value of timer T303. The appendix shows the test scenarios resulting from this approach. In this example, the AETG tool produces only a small reduction in test set size when compared to the second traditional approach. However, the AETG approach provides a much broader coverage without leaving any systematic holes in the complete test space. Moreover, the test set generated by the AETG approach is easy to implement and leads to a faster detection of non-conformances to the requirements (i.e. defects) due to a systematic coverage of the test space.

The third traditional approach would generate 17 tests: (i) The 7 cause values holding all other factors constant, (ii) the 6 bearer capabilities holding all other factors constant, and (iii) the remaining 4 as in the previous method. The appendix shows the scenarios resulting from this approach. The quality of coverage of the test space is very poor.

Thus, the third traditional approach would provide inadequate coverage and the first traditional approach would be very expensive, while adding little additional insight (compared to the AETG design) as to the behavior of the switch under test.

4.2 Channel Negotiation

Channel negotiation is a concept that applies to the terminating PRI [1]. The called Class II equipment is allowed to negotiate with the switch over which B-channel is used to terminate the incoming call. The incoming SETUP message contains a channel identification information element indicating the B-channel for which the call is intended. The called Class II equipment has the option to either agree on the switch selected B-channel or choose another available channel.

When unrestricted channel negotiation procedures apply and negotiation is unsuccessful, the following factors play a role in the treatment applied to the involved parties:

- Channel Negotiation Message (3),
- Bearer Capability (6),
- Whether the negotiation occurred before or after timer T303 has expired (2),
- Number of DS-1 facilities on the terminating interface and how the B-channel was signaled (3).

Thus, to test all combinations of the above factors requires 108 (3*6*3*2) tests. The test design obtained by applying the AETG tool provides coverage of all pairwise combinations in 18 tests. Here again, the initial design provided by the AETG tool was improved to achieve better balance. The improved design shown in Table 3 contains 17% of all possible tests.

Table 2
Call Rejection Scenarios

Cause	Originator/Bearer Capability	Call Clearing Message	CALL PROCeeding	T303
88	ISDN/Speech	DISConnect	Sent	Before
88	ISDN/3.1-kHz Audio	RELease COMPlete	Not Sent	Before
88	ISDN/56-kbps Data	RELease	Not Sent	After
88	ISDN/64-kbps Data	DISConnect	Not Sent	Before
88	Non-ISDN Line	RELease COMPlete	Sent	After
88	Non-ISDN Trunk	RELease COMPlete	Sent	After
17	ISDN/Speech	RELease	Sent	Before
17	ISDN/3.1-kHz Audio	RELease	Not Sent	After
17	ISDN/56-kbps Data	DISConnect	Not Sent	After
17	ISDN/65-kbps Data	RELease COMPlete	Sent	After
17	Non-ISDN Line	RELease COMPlete	Not Sent	Before
17	Non-ISDN Trunk	DISConnect	Sent	After
34	ISDN/Speech	DISConnect	Sent	After
34	ISDN/3.1-kHz Audio	RELease	Not Sent	Before
34	ISDN/56-kbps Data	DISConnect	Not Sent	Before
34	ISDN/64-kbps Data	RELease COMPlete	Not Sent	After
34	Non-ISDN Line	RELease	Sent	Before
34	Non-ISDN Trunk	RELease COMPlete	Sent	After
65	ISDN/Speech	DISConnect	Not Sent	Before
65	ISDN/3.1-kHz Audio	RELease COMPlete	Sent	Before
65	ISDN/56-kbps Data	RELease	Sent	Before
65	ISDN/64-kbps Data	RELease	Not Sent	Before
65	Non-ISDN Line	DISConnect	Not Sent	After
65	Non-ISDN Trunk	RELease COMPlete	Not Sent	After
27	ISDN/Speech	RELease	Not Sent	After
27	ISDN/3.1-k	RELease	Not Sent	After
27	ISDN/56-kbps Data	RELease COMPlete	Not Sent	Before
27	ISDN/64-kbps Data	DISConnect	Not Sent	After
27	Non-ISDN Line	RELease	Sent	Before
27	Non-ISDN Trunk	RELease COMPlete	Sent	Before
None	ISDN/speech	RELease COMPlete	Sent	Before
None	ISDN/3.1-k	DISConnect	Not Sent	After
None	ISDN/56-kbps Data	RELease COMPlete	Sent	After
None	ISDN/64-kbps Data	RELease	Sent	Before
None	Non-ISDN Line	RELease	Sent	Before
None	Non-ISDN Trunk	DISConnect	Not Sent	Before
Other	ISDN/Speech	DISConnect	Sent	After
Other	ISDN/3.1-kHz Audio	RELease	Sent	Before
Other	ISDN/56-kbps Data	RELease COMPlete	Not Sent	After
Other	ISDN/64-kbps Data	DISConnect	Sent	Before
Other	Non-ISDN Line	DISConnect	Sent	After
Other	Non-ISDN Trunk	RELease	Not Sent	Before

Table 3
Channel Negotiation Scenarios

Channel Negotiation Message	Originator/Bearer Capability	T303	Terminating Interface Configuration
CALL PROCeeding	ISDN/Speech	Before	Single DS-1
CALL PROCeeding	ISDN/3.1-kHz Audio	After	Single DS-1
CALL PROCeeding	ISDN/56-kbps Data	Before	Multiple DS-1; Explicit Signaling
CALL PROCeeding	ISDN/64-kbps Data	Before	Multiple DS-1; Explicit Signaling
CALL PROCeeding	Non-ISDN Line	After	Multiple DS-1; Implicit Signaling
CALL PROCeeding	Non-ISDN Trunk	After	Multiple DS-1; Implicit Signaling
ALERTing	ISDN/Speech	After	Multiple DS-1; Explicit Signaling
ALERTing	ISDN/3.1-kHz Audio	After	Multiple DS-1; Explicit Signaling
ALERTing	ISDN/56-kbps Data	After	Multiple DS-1; Implicit Signaling
ALERTing	ISDN/64-kbps Data	Before	Multiple DS-1; Implicit Signaling
ALERTing	Non-ISDN Line	Before	Single DS-1
ALERTing	Non-ISDN Trunk	Before	Single DS-1
CONNect	ISDN/Speech	After	Multiple DS-1; Implicit Signaling
CONNect	ISDN/3.1-kHz Audio	Before	Multiple DS-1; Implicit Signaling
CONNect	ISDN/56-kbps Data	After	Single DS-1
CONNect	ISDN/64-kbps Data	After	Single DS-1
CONNect	Non-ISDN Line	Before	Multiple DS-1; Explicit Signaling
CONNect	Non-ISDN Trunk	Before	Multiple DS-1; Explicit Signaling

5. Assessment of Test Coverage

In the two applications discussed here, the AETG tool has yielded test designs where the fraction of all possible test combinations covered in the test designs, n/N, is in the range 0.08 - 0.17. This fraction decreases as K, the number of factors in the application, increases. For example, if an application has $K = 7$ and $L_1 = L_2 = L_3 = 7$, $L_4 = L_5 = 3$, and $L_6 = L_7 = 2$, then the AETG design will contain 54 tests out of a total of 12348 possible tests. The fraction of all test combinations included in this test design is $n/N = 0.004$, which corresponds to a 99.6% decrease in the number of all possible tests.

On the other hand, if $K = 2$, we have $n = N$ and the test design contains all possible tests, because we require all pairwise combinations to be included. If $K = 3$ and $L_1 >= L_2 >= L_3$, then we have $n/N = 1/L_3$. Thus the potential for reduction in n/N gets larger as K increases and the benefit from the application of the AETG tool increases. The value added by applying the AETG tool is that it provides a wider breadth of coverage and approximate balancing of the various pairs of test factors without increasing the size of a typical traditional approach.

In the future, the AETG tool will be applied to more applications beyond those discussed in this paper. Also, the process of enhancing the initial test set design to improve the balance with respect to the levels of different factors in the test design will be automated.

6. Conclusion

We have discussed two applications for ISDN protocol conformance testing in which the AETG tool has proven successful at improving both the quality and efficiency of the testing process. The sizes of the test sets for these two applications based on the AETG tool are comparable to those for the second traditional approach. However, the AETG approach provides a much broader coverage of the test space without leaving any systematic holes in the complete space of all test combinations. Moreover, the test set generated by the AETG approach is easy to implement in automated test systems. The improved quality of testing leads to a faster detection of non-conformances, and a higher quality of products in a shorter development interval.

We have also identified the characteristics of other protocol applications for which the AETG tool would produce test sizes that are significantly smaller than those for the second traditional approach and at the same time yield a significant improvement in the breadth of coverage. Thus, the AETG approach described here can be applied to other types of testing (such as feature testing and operability testing between two different network elements), resulting in improved quality of testing without increasing the test set size.

7. Acknowledgment

We would like to thank Siddhartha Dalal for developing and providing us with the AETG tool. This tool provided us an initial set of test cases for the three applications discussed here. In addition, we would like to thank Roshan Chaddha and Andy Fountoukidis for providing the original motivation for our work in this area.

References

[1] Bellcore Technical Reference, TR-NWT-001268, Issue 1, December 1991, ISDN PRI Call Control Switching and Signaling Generic Requirement for Class II Equipment.

[2] Taguchi, G. (1991), System of Experimental Design, Quality Resources and American Institute, Inc.

[3] Phadke, M. S. (1989), Quality Engineering Using Robust Design, Prentice Hall.

[4]. Perkinson, W. B.(1992), A Methodology for Designing and Executing ISDN Feature Tests, Using Automated Test Systems, IEEE GLOBECOM'92 Conference Record.

APPENDIX

(i) Call Rejection: Third Traditional Approach

Cause	Originator/Bearer Capability	Call Clearing Message	CALL PROCeeding	T303
88	ISDN/Speech	RELease COMPlete	Sent	Before
17	ISDN/Speech	RELease COMPlete	Sent	Before
34	ISDN/Speech	RELease COMPlete	Sent	Before
65	ISDN/Speech	RELease COMPlete	Sent	Before
27	ISDN/Speech	RELease COMPlete	Sent	Before
None	ISDN/Speech	RELease COMPlete	Sent	Before
Other	ISDN/Speech	RELease COMPlete	Sent	Before
88	ISDN/Speech	RELease COMPlete	Sent	Before
88	ISDN/3.1-kHz Audio	RELease COMPlete	Sent	Before
88	ISDN/56-kbps Data	RELease COMPlete	Sent	Before
88	ISDN/65-kbps Data	RELease COMPlete	Sent	Before
88	Non-ISDN Line	RELease COMPlete	Sent	Before
88	Non-ISDN Trunk	RELease COMPlete	Sent	Before
88	ISDN/Speech	DISConnect	Sent	Before
88	ISDN/Speech	RELease	Sent	Before
88	ISDN/Speech	RELease COMPlete	Not Sent	Before
88	ISDN/Speech	RELease COMPlete	Sent	After

APPENDIX (Continued)

(ii) Call Rejection: Second Traditional Approach

Cause	Originator/Bearer Capability	Call Clearing Message	CALL PROCeeding	T303
88	ISDN/Speech	RELease COMPlete	Sent	Before
88	ISDN/3.1-kHz Audio	RELease COMPlete	Sent	Before
88	ISDN/56-kbps Data	RELease COMPlete	Sent	Before
88	ISDN/64-kbps Data	RELease COMPlete	Sent	Before
88	Non-ISDN Line	RELease COMPlete	Sent	Before
88	Non-ISDN Trunk	RELease COMPlete	Sent	Before
17	ISDN/Speech	RELease COMPlete	Sent	Before
17	ISDN/3.1-kHz Audio	RELease COMPlete	Sent	Before
17	ISDN/56-kbps Data	RELease COMPlete	Sent	Before
17	ISDN/65-kbps Data	RELease COMPlete	Sent	Before
17	Non-ISDN Line	RELease COMPlete	Sent	Before
17	Non-ISDN Trunk	RELease COMPlete	Sent	Before
34	ISDN/Speech	RELease COMPlete	Sent	Before
34	ISDN/3.1-kHz Audio	RELease COMPlete	Sent	Before
34	ISDN/56-kbps Data	RELease COMPlete	Sent	Before
34	ISDN/64-kbps Data	RELease COMPlete	Sent	Before
34	Non-ISDN Line	RELease COMPlete	Sent	Before
34	Non-ISDN Trunk	RELease COMPlete	Sent	Before
65	ISDN/Speech	RELease COMPlete	Sent	Before
65	ISDN/3.1-kHz Audio	RELease COMPlete	Sent	Before
65	ISDN/56-kbps Data	RELease COMPlete	Sent	Before
65	ISDN/64-kbps Data	RELease COMPlete	Sent	Before
65	Non-ISDN Line	RELease COMPlete	Sent	Before
65	Non-ISDN Trunk	RELease COMPlete	Sent	Before
27	ISDN/Speech	RELease COMPlete	Sent	Before
27	ISDN/3.1-k	RELease COMPlete	Sent	Before
27	ISDN/56-kbps Data	RELease COMPlete	Sent	Before
27	ISDN/64-kbps Data	RELease COMPlete	Sent	Before
27	Non-ISDN Line	RELease COMPlete	Sent	Before
27	Non-ISDN Trunk	RELease COMPlete	Sent	Before
None	ISDN/speech	RELease COMPlete	Sent	Before
None	ISDN/3.1-k	RELease COMPlete	Sent	Before
None	ISDN/56-kbps Data	RELease COMPlete	Sent	Before
None	ISDN/64-kbps Data	RELease COMPlete	Sent	Before
None	Non-ISDN Line	RELease COMPlete	Sent	Before
None	Non-ISDN Trunk	RELease COMPlete	Sent	Before
Other	ISDN/Speech	RELease COMPlete	Sent	Before
Other	ISDN/3.1-kHz Audio	RELease COMPlete	Sent	Before
Other	ISDN/56-kbps Data	RELease COMPlete	Sent	Before
Other	ISDN/64-kbps Data	RELease COMPlete	Sent	Before
Other	Non-ISDN Line	RELease COMPlete	Sent	Before
Other	Non-ISDN Trunk	RELease COMPlete	Sent	Before
88	ISDN/Speech	DISConnect	Sent	Before
88	ISDN/Speech	RELease	Sent	Before
88	ISDN/Speech	RELease COMPlete	Not Sent	Before
88	ISDN/Speech	RELease COMPlete	Sent	After