Copyright © 1979 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 58, No. 6, July–August 1979
Printed in U.S.A.

Traffic Service Position System No. 1:

Remote Trunking Arrangement: Hardware and Software Implementation

By A. F. BULFER, W. E. GIBBONS, and J. A. HACKETT

(Manuscript received December 28, 1978)

The principal data-handling circuits used between a TSPS base location and remote sites up to 1000 miles distant are described. Included are novel operational programming and fault recognition features required by this large separation between the processor and its peripherals as well as by the uniqueness of certain remote units. The RTA's use of remotely run diagnostics and multi-unit initialization schemes for fast recovery from outages is also covered.

I. INTRODUCTION

Early in the RTA planning process, it was recognized that controlling trunks, networks, and test circuits remotely was a task similar to controlling operator positions, voice path circuits, and test circuits remotely; i.e., the RTA job would be similar, in several aspects, to the work planned for development of a new position subsystem. These configurations are shown in Fig. 1. These job similarities led to development of a set of data-handling circuits known as the Peripheral Control Link (PCL) used to link the remote hardware to the controlling processor at a TSPS base location. A Remote Trunk Arrangement (RTA) consists of a PCL, various 2-wire and 4-wire trunks at the remote site, a concentrator network to temporarily connect these remote trunks to a base-remote trunk, a maintenance buffer, and a trunk test frame. A Position Subsystem (PSS) No. 2 consists of a PCL, several operator positions, some supervisory and chief operator circuits, a voice path control frame, a maintenance buffer, and a test frame.

The PSS No. 2 is only briefly treated in this paper and is mentioned at this time to indicate another consideration in PCL design—a desire

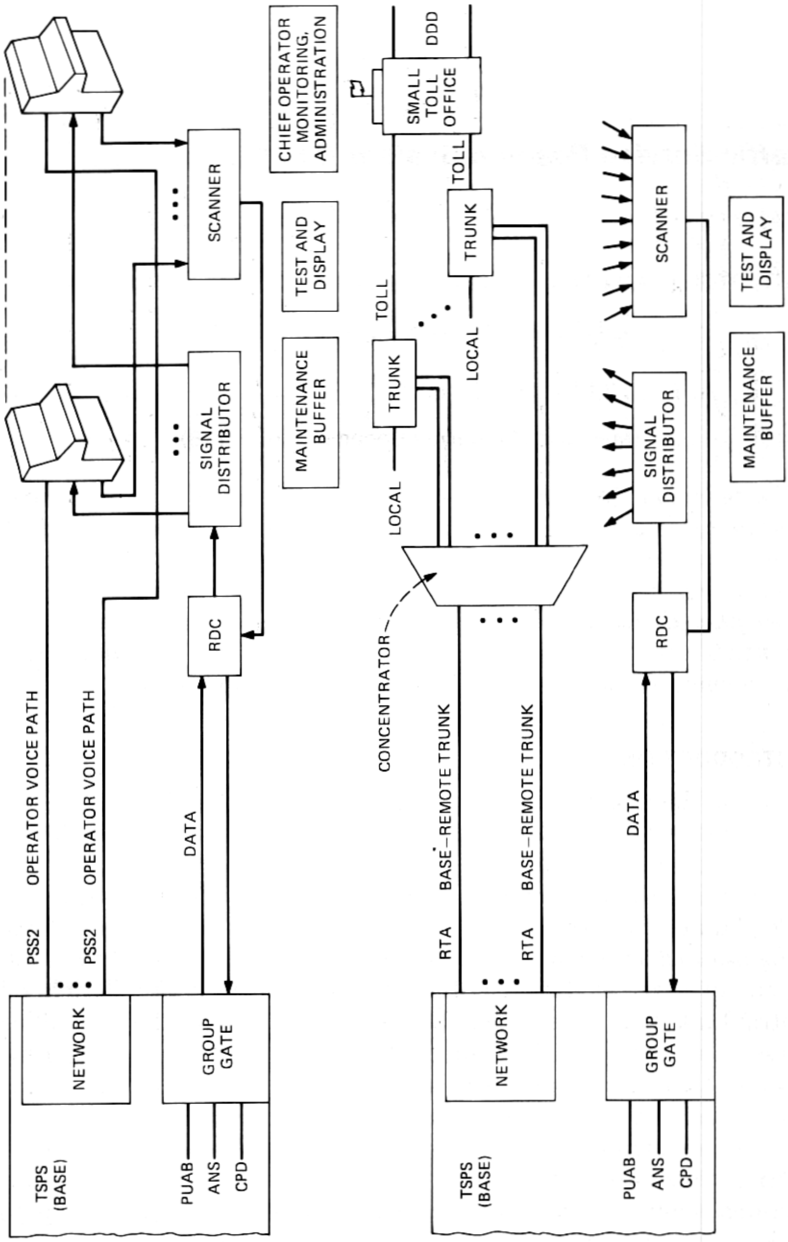

Fig. 1—RTA/PSS No. 2 common equipment.

to keep the PCL general purpose and to have its data rate high enough to handle bursts of operator activity. In both the RTA and the PSS No. 2 configurations, the remote circuits (trunks or positions) are temporarily connected to the base TSPS network in processing a call. In the RTA case, a base-remote trunk is used so that various "service" circuits at the TSPS base location (digit receivers, outpulsers, tones, announcements, etc.) can be connected to the RTA trunk as needed. In the PSS No. 2 case, operators at remote positions are temporarily connected through the base TSPS network to customers placing toll calls requiring operator assistance. The many voice circuits required in each case (up to 64 base-remote trunks or operator voice paths) utilize carrier transfer circuits and carrier failure detection circuits to permit fast, processor-controlled switching to spare carrier groups in the event of trouble. This carrier group switching is similar to that used in other switching designs and is only briefly discussed in Section 2.5.

The PCL includes a base location data conversion circuit known as a group gate, outside plant facilities, remote data circuit, signal distributor, scanner, and diagnostic control circuit. Each of these circuits consists of two identical halves and are arranged in two separate PCL halves with triplicated transmission facilities as shown in Fig. 2. A PCL half is capable of sustaining all communications and control between the Stored Program Control (SPC) and the RTA or PSS No. 2 circuits. When both PCL halves are in service, a state referred to as "duplex,"

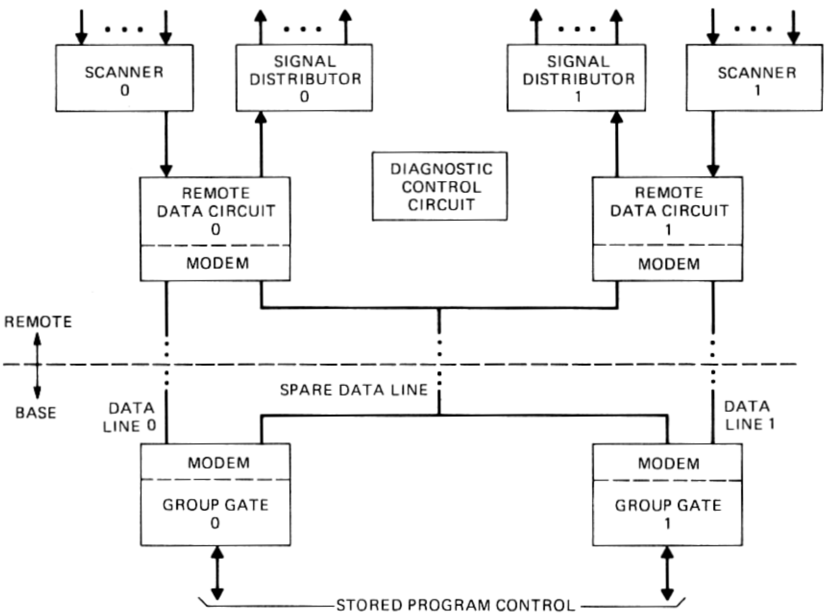

Fig. 2—Peripheral control link.

the PCL throughput remains the same, but the error detecting capabilities of the PCL are increased. Several reliability studies[1-3] have shown the transmission paths to be the weakest link and other studies have shown matching to be one of the best methods of error detection. This led to use of triplicated facilities so that duplex operation *with matching* can be continued even with a facility outage. Operation without matching is also possible so that, while error-detecting capability is lessened, complete RTA or PSS No. 2 operation can continue with two simultaneous facility outages. When a PCL half is out of service, the PCL is said to be in a simplex state.

The RTA equipment not considered part of the PCL is not duplicated. However, control of this part of the RTA is distributed over a variety of individual control devices, such as trunk buffers, concentrator controllers, and maintenance buffers. As a result, failure of any one control device affects only a small part of the RTA complex. For example, a trunk buffer failure would affect no more than 16 trunks.

II. HARDWARE AND PHYSICAL DESIGN

The principal parts of the RTA hardware are the PCL, the concentrator, and the trunks. Each of these parts is described in this section. A test frame, including a 2-way TTY for maintenance messages, is also provided and is described briefly in Section 4.5. The principal parts of the PSS No. 2 hardware are the PCL and the 100C operator positions, and the 100C positions are also described in this section.

2.1 PCL description

The PCL is a general-purpose data link designed to automatically retry failing data words so that fault recognition programs will not be utilized for minor transmission errors. Four principal checks are made on each direction of transmission: parity, cyclic code, sequence check, and matching. In general, the data to be sent are applied to both PCL halves and treated independently by each half. Then, at the far end of the PCL, the data from both halves are matched, and a control bit is examined to see which PCL half will actually execute the order. Certain combinations of matching failures, cyclic code failures, sequence failures, and parity failures still permit valid execution of the transmitted orders, and these possibilities are discussed in Section 4.1.

2.1.1 Group Gate and remote data circuit

The Group Gate (GG) connects the PCL to the address bus, answer bus, and Central Pulse Distributor (CPD) of the TSPS processor. A Group Gate frame provides up to four fully duplicated Group Gates and the common bus input/output circuits to send and receive data to all four. This common bus circuitry initially accepts a 21-bit binary word from the address bus and passes it to all Group Gates on the frame. An enable signal from the CPD then designates which Group Gate should respond to this particular 21-bit word, i.e., there is a separate enable signal for each Group Gate on the frame. There is also a common enable designating whether this order is "odd" or "even," and circuitry at the remote end continually checks that an alternating pattern of odd and even words is received.

The Group Gate also functions as a parallel-to-serial converter (in the sending direction) and thus converts the 21-bit parallel data into a 21-bit serial bitstream. The odd/even bit mentioned above is added, and a 2-bit "start" code is also added to designate the start of a new transmission and also to designate whether the new word is a long word (*data* from the TSPS processor) or a short word (*scan complete* signal from the processor). The resulting 24-bit serial word (long word) is then passed through a cyclic code generator which appends a 5-bit cyclic code. The resulting 29-bit serial word is fed to a 2400-b/s data set which performs a digital-to-analog conversion and connects to the transmission facility. (The "short word"—an acknowledgment signal from the processor that the last transmission from the distant end was received correctly—is 11 bits long as it is passed to the data set.) In normal (duplex) operation, both halves of a Group Gate receive the 21-bit word from the address bus and generate the 29-bit analog pulse streams to the two transmission paths. However, it is also possible to split the PCL into two (simplex) halves, and in this mode each half can be sent a different order from the processor.

In the receiving direction (Fig. 3), each Group Gate half receives an analog pulse stream into the 2400-b/s data set where it is converted to a digital bitstream. This is passed to a cyclic code check circuit and a parity check circuit. A processor-controlled activity bit designates one Group Gate half as "active" and the other as "inactive." Both halves

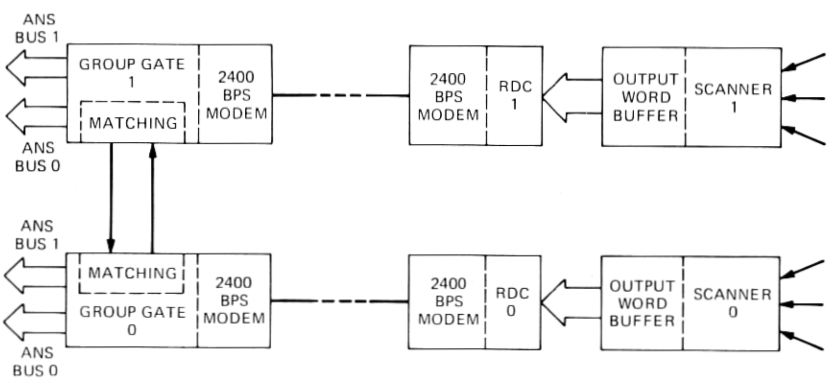

Fig. 3—PCL receive chain.

perform all the actions outlined above, but only the active half controls the matching operation. After the cyclic code and parity checks are performed, the active half moves data from both halves through a serial-mode matching circuit in the active half and determines if the match is valid or invalid. If cyclic code and parity tests have passed and matching is valid, then a signal is sent to the TSPS processor that the Group Gate has information ready to be passed on. If only the active half passed the cyclic code and parity checks and matching with the inactive half is *not* valid, then, again, the processor is signaled that information is present. A CPD pulse from the processor then causes that Group Gate to send its information on the processor answer bus.

For other combinations of invalid data and for transmission "burst" errors[2] affecting more than 1 bit and detected by the cyclic code check, a signal is sent to the remote end to retransmit the data. The processor is also informed that a retransmission or several retransmissions are taking place and, depending upon the severity of the problem, certain fault recognition routines may be executed. For example, the error activity might indicate a bad transmission facility on one PCL half, and the corrective action would be to switch to a spare transmission facility, or the error might be a simple transmission "hit" or "burst" which is cleared up by a retransmission, and no other maintenance activity is required.

The Group Gate can also receive processor data into a maintenance register which sets up any of several specialized maintenance states within a Group Gate half. For example, the cyclic code generator can be configured to purposely output bad codes so that subsequent PCL circuits can be exercised. The sending portion of a Group Gate half can also be connected back to the receiving portion of the same Group Gate half so that data from the TSPS processor can be operated upon by the Group Gate and returned to the processor as part of a diagnostic routine.

The Remote Data Circuit (RDC) is very similar to a Group Gate and, in fact, uses many of the same circuit packs designed for the Group Gate. However, it is located at the remote end of the PCL and connects to the signal distributor and the scanner. Like the Group Gate, it performs parallel-to-serial and serial-to-parallel conversions, parity checks, cyclic code generation and checking, odd/even sequence checking, and matching. The remote data handling circuits are mounted on a double-bay frame such that one side of the frame contains all the 0 half circuits and the other bay contains all the 1 half circuits. The two RDC halves, including their respective 2400-b/s data sets, each comprise one 6-in. mounting plate on the two bays of the frame. This frame, called the position and trunk control frame (for use both in PSS No. 2 and in RTA) is shown in Fig. 4.

Whenever no new data transmissions are available from either the

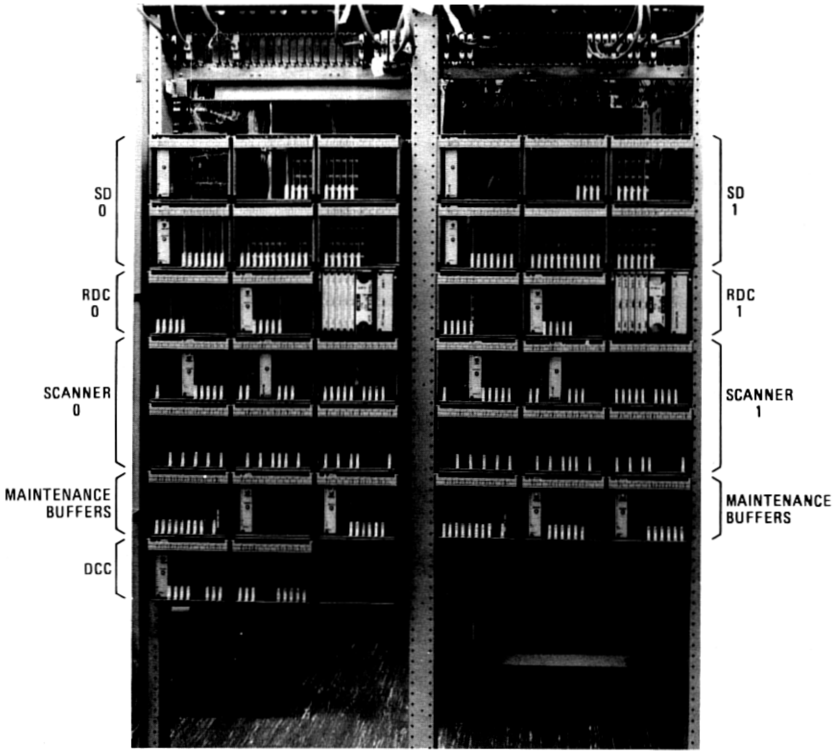

Fig. 4—Position and trunk control frame.

processor (for the Group Gate) or from the remote scanner (for the RDC), a dummy word is sent over the PCL. Thus, the PCL is sending information back and forth every 25 ms even during inactive periods. Both the Group Gate and RDC have the ability to recognize nonreceipt of such interchanges as a mechanism for identifying transmission facility troubles. In addition, the RDC, after several seconds of not receiving any words (dummy or data) from the Group Gate, will split the remote PCL end into two simplex halves. This action is based on the assumption of a trouble which has been recognized at the base (processor) location and has somehow prevented orders (to switch to simplex) from being executed at the remote location. The automatic switch to simplex operation then allows the fault recognition programs to exercise both halves and all three transmission facilities so as to find a good configuration.

2.1.2 Position and Trunk Scanner

The scanner is considered a part of the PCL, but introduces the concept of distributed control, since much of it is physically mounted on the trunk frames and 100C consoles with which it works. This distributed concept means that only that portion required need be equipped, i.e., as a new trunk frame is added to an office, another piece of the scanner is essentially added to the scanner. This new piece of the scanner was furnished as a part of the trunk frame just added and deals with supervisory signals from those added trunks. It is then connected to the rest of the scanner which is provided as two independent halves, each reporting changes of state to the processor at the TSPS base location. At the base location (in the Group Gates, as described in Section 2.1.1), the two scanner reports are matched before actual transmission to the processor. Mismatch errors are a prime means for detecting scanner malfunctions (see Section 4.1 below).

The scanner is an autonomous circuit continually looking for transitions such as changes in trunk supervision, depression of a key on a position, or test frame and system alarms. It is driven by a clock signal obtained from the 2400-b/s data set used by the RDC and both scanners are run from a single data set clock. However, this clock signal is continually checked and both scanners will switch to the other data set (RDC) clock in case of trouble. For simplex operation, the two scanners each connect to the corresponding individual RDC clocks. Minute differences in the two RDC clocks require a period of synchronization as two simplex scanner halves are reconfigured into a single duplex system.

The scanner examines some 1300 scan points every 12.3 ms and records the state of each in Random Access Memory (RAM). In the *next* 12.3-ms cycle, the previously recorded information is known as the "Last Look" (LL). At any instant of time, the current state of a scan point on which the scanner is currently acting is known as the "Present Look" (PL). The scanner also keeps track (in RAM) of the previously *reported* state of each scan point. This is known as the "Previous State" (PS) and is the state last reported to the TSPS processor. As the scanner examines each point, it compares the PL to the LL and PS. If the PL and LL agree *and* are different from the PS, then a report is made and PS is changed to the new value. The requirement that two scans spaced 12.3 ms apart must agree allows the scanner to disregard all transient conditions producing pulses shorter than 12.3 ms and guards against line and supervisory transients being reported as valid trunk states.

Although both scanners are running on a single clock during duplex operation, there is a low probability chance that one scanner half will see a scan point transition one scan interval ahead of the other. Then, on the next scan, one scanner will make a report while the other scanner has seen the transition for the first time and will not make a report. On the third scan, the second scanner would make its report and from that point on the internal scanner states should be in agreement. But in the meantime, a mismatch at the Group Gate may have been detected and some fault recognition activity would be triggered needlessly since the mismatch does not indicate a hard fault. To prevent this transient mismatch, a match on the present look between the scanners is performed. If a mismatch occurs at this point, then both scanners ignore the present look in this scan period. On the second scan, if the mismatch was not a hard fault, both scanners will agree and a report will be made in the normal fashion. If the mismatch persists through the second scan, both scanners are again allowed to proceed in a normal fashion but in this case a mismatch will be generated and the hard fault will be detected. A single bit per scan point is required to count the present look mismatches (PLMM) and this bit is kept in RAM along with the LL, PS, and a parity bit for RAM self-checking.

The scanner presents reports to the RDC which adds cyclic code and converts to a serial analog pulse stream. The RDC accepts the scanner reports at a maximum rate of one every 25 ms, depending upon data line activity. However, the scanner *input* rate is based on whatever activity is happening at some interval of time and could conceivably be much greater than a 25-ms rate for brief periods of time, i.e., bursts of operator activity whereby several operators simultaneously push keys or RTA trunk activity involving dial pulse reception on several trunks. Also, the 25-ms output rate may be slowed due to transmission line errors which require retransmissions. To cope with such activity, the scanner contains a 32-word output buffer so that reports can be stored temporarily until they are transmitted. Calculations indicate that, even with bursts of operator and trunk activity, a 32-word buffer will almost always guarantee that no information will be lost.[4] Exceptions can occur during severe (but very infrequent) maintenance activity, when scanner reports accumulate while fault recognition is running maintenance tests under partial outage conditions.

A feature of the scanner is its ability to count dial pulses on the incoming (local office) side of every trunk, recognize the interdigit interval, and report dialed *digits* to the TSPS processor. It also performs timing checks on initial state changes from the local office side of a trunk to recognize "false seizure" conditions so as to not report such false seizures (these can persist for up to 40 ms). These features are based on defining several possible trunk states such as initial seizure, start of open interval of a dial pulse, start of closed interval of a dial pulse, start of interdigital interval, end of pulse, end of digit, disconnect, etc. A trunk progresses from state to state by timing the number of 12.3-ms cycles in which it remains in an on-hook or off-hook condition. Thus, each time the trunk condition changes, a count of 12.3-ms cycles is started (by incrementing a 4-bit count—4 RAM bits—assigned to the incoming side of each trunk). The actual count reached before the on-hook/off-hook condition changes determines the next state of the trunk. For example, consider a trunk receiving dial pulses which, after a few pulses of a digit, starts counting the 12.3-ms cycles of an off-hook interval. If 2 to 10 cycles are counted and then the trunk condition changes to on-hook, this is considered another pulse of the digit, and the on-hook interval is timed to confirm this. However, if 11 or more (off-hook) 12.3-ms cycles are counted, this is considered an interdigital interval and the previously counted pulses now represent the digit to be reported.

The items scanned by the scanner include trunk supervisory inputs, 100C console keys, and alarm indications from all remote circuits. The reported information consists of scan point type, unit number, and the particular status change of that unit (such as operation of the START TIMING key at a position). A parity bit and a sequence number are added by the scanner. The sequence number is a continuously incremented 2-bit binary number (0-1-2-3-0-1-· · ·) indicating the order in which reports are loaded by the scanner into its 32-word output buffer. The RDC adds a start code, a 5-bit cyclic code as described for the Group Gate in Section 2.1.1, and sends the report back to the TSPS processor. At the processor, this sequence code from the scanner is continually checked and any out-of-sequence condition is a trigger for fault recognition action. This guards against a report being lost with no indication to the processor that the lost report ever existed.

The scanner is constantly performing self-checks as it scans across certain scan points reserved for maintenance self-checks. For example, one scan point is arranged to give on-hook/off-hook sequences that should represent a certain digit count and other circuitry within the scanner checks that this count has been reached. These checks, in addition to parity checks across the RAM, influence an "all-seems-well" bit returned with each scanner report and are constantly examined by the TSPS processor. A failing "all-seems-well" bit triggers immediate fault recognition action.

2.1.3 Signal distributor

The signal distributor takes parallel information from the RDC, strips off the unit designation and unit type information (such as position number, trunk buffer number, concentrator controller number, etc.), and distributes the remaining portion of the order to the designated unit. In performing this distribution, it converts the parallel output of the RDC to a serial bipolar bitstream and adds parity and start bits. The resulting serial word is transmitted over a single pair of wires to an RTA trunk buffer, a 100C console (PSS No. 2), a concentrator controller, a maintenance buffer, or to the test frame. Thus the signal distributor communicates with 70 to 80 units of five different types.

(For a PSS No. 2, this consists of 62 positions, 5 maintenance buffers, 2 test frame buffers, and 1 self-test buffer. For an RTA, this consists of 31 trunk buffers, 5 maintenance buffers, 2 test frame buffers, 40 concentrator controllers, and 1 self-test output). The serial bit-bipolar transmission allows these units to be several hundred cable feet from the signal distributor.

After the serial word is constructed within the signal distributor and loaded into an output shift register, it is then shifted out and, at the same time, looped back into the output register. Thus, when shifting (transmission) is complete, the transmitted word is again in the output register and is rechecked bit by bit for accuracy. Both this check and a subsequent flip/flop activity check made by the receiving unit (100C console, trunk buffer, etc.) influence the check-back signal sent from the signal distributor back to the RDC, and subsequently to the SPC, as an overall check. This shifting action also allows calculation of a parity bit "on-the-fly," i.e., as the information and start bits are shifted out, parity is calculated and, if the shift register recheck passes, the correct parity bit is appended to the end of the serial transmission. Thus the parity bit can be purposely "failed" to keep the receiving unit from using questionable data.

The start code not only identifies the start of a new transmission but also identifies the *type* of unit expected to receive the new information. Each receiving unit contains a start code mask that must match the start code to accept the information bits. Each receiving unit (buffer) also generates either a "check-back" or a "buffer failure" response to the signal distributor each time it receives new information. This response is returned to the distributor over the same pair of wires used to send information to a buffer. This response, together with the signal distributor's internal checks, is used to send a check-back signal to the RDC and from there to the SPC. If the SPC fails to receive this check-back, fault recognition action is triggered, starting with a simple retry and escalating to subsystem reconfigurations if necessary.

The duplex nature of the sending portion of the PCL effectively ends at the RDC and, once RDC matching is complete, only one of the duplicated signal distributors executes the order. The distributor chosen by the RDC is changed on every order to utilize both distributors evenly. If the PCL is split into two independent halves for maintenance purposes, then maintenance orders can use one signal distributor while call processing orders use the other.

The orders to light LED numerical displays in 100C consoles place special requirements on the signal distributor. These orders are always sent as a sequence of 2 to 7 consecutive SPC orders, but only the first contains the address of the desired 100C console. The signal distributor recognizes such orders (usually called "prime" orders since they prime the distributor and the 100C console to expect a sequence) and temporarily stores the position number while passing the rest of the order to the desired position. Subsequent SPC orders will contain pairs of digits to be selected on the console LED display but do not include a position number. Since only one signal distributor executed the "prime" order and remembers the position number, that one distributor handles all of the rest of the sequence and the alternating use of both signal distributors is temporarily suspended.

The "prime" order also informs the distributor as to which of several *types* of LED displays will be forthcoming; e.g., a 12-digit (including blanks) display of a called number, a 1-digit display of elapsed minutes, a 6-digit display of the time-of-day, etc. The distributor includes a 6-unit ring counter that chooses the location, within the operator's numerical display, where the pair of digits just received are to be placed. This ring counter is preloaded by the prime order to start the first pair of digits (next SPC order) in the correct position on the 100C console. The ring counter is then stepped along by each subsequent order so that it is always ready to direct the digits to the proper place. As the signal distributor forms the 23-bit serial word to the 100C console it takes the digit pair *values* from the RDC, the pair *location* from the ring counter, and the 100C *address* from its stored memory of that address, and adds a parity bit computed on-the-fly as the other information goes by.

2.1.4 Diagnostic control circuit

The diagnostic control circuit (DCC) is also considered part of the PCL as shown in Fig. 2. It allows implementation of a remotely run diagnostic concept explained in Section 4.3. The overall description, together with its control by the SPC and its considerable access to remote PCL circuits, are all included in Section 4.3 as part of the RTA maintenance plan.

2.2 Concentrator

2.2.1 Size, ratio, components

The RTA concept is to provide TSPS services for a sparsely populated area from a TSPS in a more built-up area. Therefore, many RTA installations consist of 200 to 400 trunks arranged in many small trunk groups with relatively low occupancy. Initially, only 100 to 150 trunks may be required, with growth to 300 to 400 taking place over several years. A few RTA sites will have larger trunk groups and correspondingly higher occupancy and a few will have trunks exceeding the capacity of a single RTA and use two RTAs in the same building.

Early studies of the potential RTA market and of trunk usage led to the conclusion that the base-remote trunks (from the output side of the concentrator to the controlling TSPS base installation) will be connected to a call for about one-eighth of the total holding time.[7]

Several concentration ratios were evaluated based on trunk numbers mentioned above, and the 8:1 concentrator was selected as the best compromise. The compromise was such that a slightly more expensive concentrator is initially provided but growth, on both the basic concentrator and the subsequent addition of a build-out frame, is easily accomplished. The 8:1 concentrator works well with lightly loaded offices, allows up to 496 incoming trunks, and provides for flexibility in load balancing. With a full complement of base-remote trunks (up to 64), it also works well in moderately heavily loaded offices.

The desire for a low-cost, easily-added-to, and moderately-long-holding-time switch led to use of miniature crossbar switches as the principal concentrator element. In RTA use, the select mechanism of the crossbar switches will accumulate its lifetime operations (estimated at 20 to 25 million operations) in 7 years. Ease of replacing this mechanism plus widespread operating company familiarity with such work also influenced the crossbar switch choice.

2.2.2 The logical concentrator and the physical concentrator

The logical layout of the concentrator is used to understand the connection pattern and to assign switch numbers useful to the TSPS programs controlling concentrator operations. However, the physical construction of the concentrator makes use of 20 × 10 crossbar switches arranged in a rather specialized way to achieve the 16 by 8 and 32 by 8 switches of the "logical" concentrator. This section explains the relationship between these two views of the concentrator.

The concentrator may be obtained in two sizes: a "small" size, able to connect up to 248 remote incoming trunks to as many as 64 base-remote trunks, and a "large" size, which can handle up to 496 remote incoming trunks. In each size, every remote trunk has full access to each of the up to 64 base-remote trunks. The term "remote trunk" is used broadly here to mean any trunk circuit which appears on the inlet side of the concentrator, while the term "remote incoming trunk" means specifically an RTA incoming trunk. A variety of possible remote trunks besides remote incoming trunks includes: hotel/motel auto-quote trunks, inward trunks, and test appearances. With the exception of test appearances, if any of these other types of remote trunks appear on the concentrator, they must replace remote incoming trunks so that the maximum number of the latter is reduced accordingly.

2.2.2.1 The Logical Concentrator. Figure 5 is a simplified depiction of the Logical Concentrator. In the small sizes, the first or trunk stage consists of thirty-two 16 by 8 three-wire crossbar switches, while the second or base-remote stage has eight 32 by 8 three-wire switches. The small Logical Concentrator thus has 512 inlets (2 for each of the 248 remote incoming trunks, 2 for test appearances, and 14 spares), 64 outlets (one for each of the 64 base-remote trunks), and 256 links (one

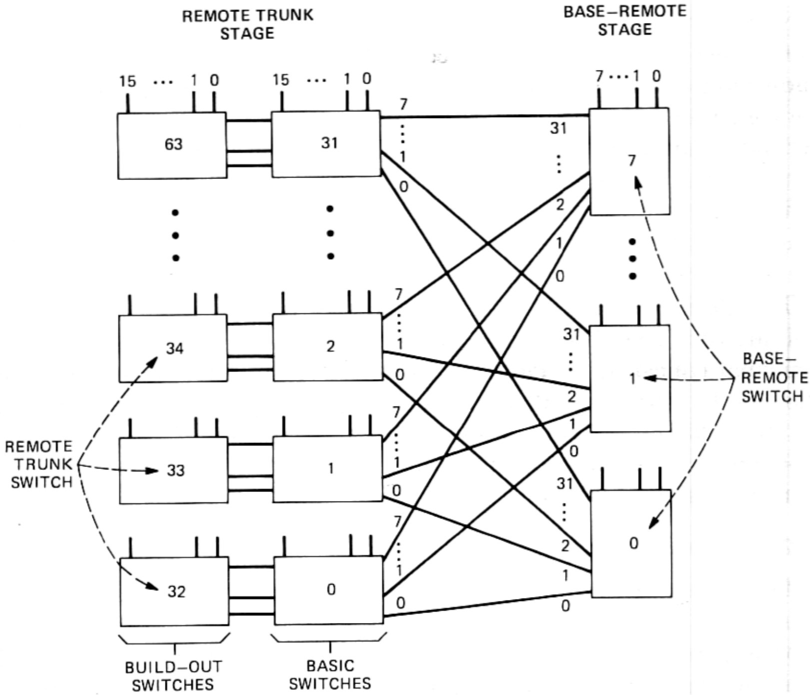

Fig. 5—The logical concentrator and its numbering.

from each of the 32 trunk stage switches to each of the 8 base-remote stage switches).

The large Logical Concentrator is obtained by adding a 16 by 8 three-wire buildout switch to each of the trunk stage switches of the small Logical Concentrator. It thus has the equivalent of thirty-two 32 by 8 trunk stage switches and eight 32 by 8 base-remote stage switches for a total of 1024 inlets (two for each of the 496 remote incoming trunks, two for test appearances, and 30 spares), 64 outlets (one per base-remote trunk), and 256 links.

It is important to note that there is only one possible path or "channel" from a *particular* remote trunk appearance to a *particular* base-remote trunk. By comparison, the base TSPS network has eight possible channels from a given inlet to a given outlet.

To make a connection in the Logical Concentrator from a particular remote trunk to a particular base-remote trunk, a *single* crosspoint is activated on the trunk stage switch containing the remote trunk and another *single* crosspoint is activated on the base-remote stage switch on which the base-remote trunk is terminated. The first crosspoint connects the remote trunk to the link going to the desired base-remote stage switch, and the second connects that link to the proper base-remote trunk.

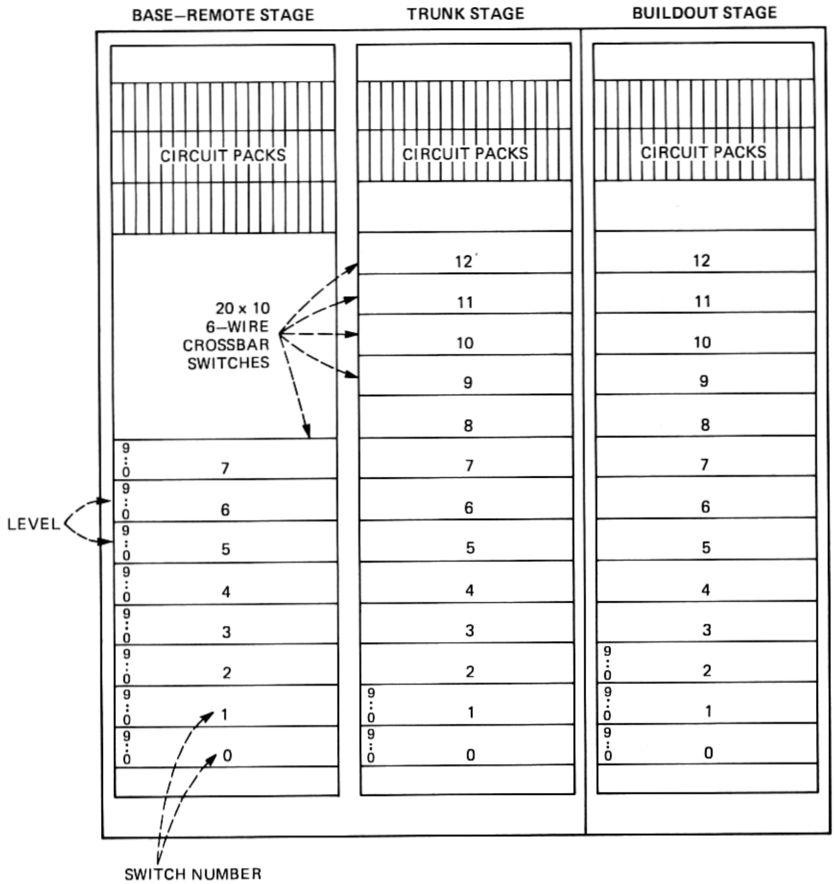

Fig. 6—The physical concentrator frames and their numbering.

2.2.2.2 The Physical Concentrator. Because of limitations on the type of crossbar switches that are available, the actual hardware implementation of the concentrator is quite different from that described above. This hardware implementation is called the Physical Concentrator and is shown in Fig. 6. The small concentrator consists of two bays of crossbar switches. The first, or trunk stage bay, contains thirteen 20 by 10 six-wire crossbar switches, while the second or base-remote stage bay contains eight 20 by 10 six-wire crossbar switches. The large concentrator is obtained by adding a second frame, similar to the trunk stage bay above, called the buildout frame which contains thirteen 20 by 10 six-wire crossbar switches.

To make the 20 by 10 six-wire switches in the physical implementation of the RTA concentrator function logically like 16 by 8 three-wire switches, a rather complex design has evolved. The first step, that of making six-wire switches behave like three-wire switches, is performed using a well-known technique[5] by which levels 0 and 1 are utilized to translate which three of the particular set of six wires are to be used. The result of this translation is that the 20 by 10 six-wire switches operate like 20 by 16 three-wire switches. These are then combined in a fairly complex way to form a unit which operates like the Logical Concentrator.

Thus, in the Physical Concentrator, to make a connection from a particular remote trunk to a particular base-remote trunk, two crosspoints must be activated in the trunk stage switch on which the remote trunk is terminated and two more crosspoints must be activated in one of the base-remote stage switches to which the base-remote trunk is connected. The first of these crosspoints is located in levels 2 to 9 and connects a pair of remote trunk appearances (six wires) to the link going to the desired base-remote stage switch. The second crosspoint is located in levels 0 or 1 and selects which of the pair is desired. The third and fourth crosspoints perform similar functions in the base-remote stage of the concentrator.

2.2.3 Distributed control

All the concentrator switch select and hold magnets are operated by controller circuits arranged on three plug-in circuit packs. The eight groups of packs used for control of base-remote stage switches are physically mounted on the concentrator frame and are always provided. However, the groups of circuit packs controlling each trunk stage logical switch (up to 32) are provided as part of the trunk buffer circuit. An RTA office can have up to 32 trunk buffers, 31 of which can each contain 16 trunks. Thus, as each group of 16 trunks is added to an RTA, a corresponding piece of the concentrator (the three controller circuit packs) is also added. The control of the trunk stage portion of the concentrator is thus distributed over all RTA trunk buffers arranged on up to five separate frames. Some cost saving is obtained by this arrangement, since only the portion of concentrator control actually required by trunks present in the office are purchased. However, of much more importance is the distribution of this control over many equipment units and the resulting overall reliability obtained; i.e., problems affecting a group of trunks affect only that part of the concentrator serving those particular trunks.

The SPC orders distributed to a particular concentrator controller by the signal distributor specify the switch number desired, the particular select and hold magnets to be used, and the function (connect or disconnect). A single trunk stage controller controls two logical switches, since each RTA trunk has two appearances on the concentrator. These two appearances (local office side and toll office side) are provided so that an outpulser connection to the toll office can be set up while an operator connection to the calling customer is established.

[Thus the 8:1 concentration ratio is approximately the ratio of incoming trunks (up to 496) to base-remote trunks (up to 64), while the ratio of trunk appearances is actually 16:1.] In general, the sequence of control orders to establish a concentrator connection is as follows: Trunk stage connect order, followed by a base-remote stage connect order, followed by operation of relays in the selected base-remote trunk. A verification test later performed by the base-remote trunk circuit results in establishment of a control path via the concentrator sleeve lead which locks the trunk stage hold magnet to the base-remote stage hold magnet. Connections are broken by a base-remote stage disconnect order which resets the base-remote stage hold magnet flip-flop and in turn releases both the base-remote and trunk stage hold magnets.

2.3 RTA trunks

The general approach to trunk design for electronic offices has been to keep the trunk simple and temporarily connect it to more sophisticated equipment for such things as digit reception, ANI reception, coin control signal generation, and outpulsing. That this requires many trunk and link orders for each call is of little concern when the trunks and their associated scanners and signal distributors are close to the processor and reached over high-speed buses. However, when the trunks are several hundred miles from the controlling processor, and when excessive numbers of orders can exceed the capacity of relatively slow-speed data paths, the simple trunk concept is not so appropriate. Thus the RTA trunks tend to be more complex than their base-location counterparts and include several features on a per-trunk basis which might otherwise be provided by a separate pool of circuits.

Another feature of each RTA trunk is that it is packaged on a single 5 by 7 in. circuit pack. The single Printed Wiring Board (PWB) construction is made possible by using miniature wire-spring relays, miniature mercury relays, miniature networks, and a specially designed integrated circuit. A typical trunk is shown in Fig. 7. The custom integrated circuit handles several signaling tasks, including coin control signals and ringback signals sent from TSPS to the local office.

The coin control and ringback considerations, including the ability to disable or enable a station set *TOUCH-TONE*® dialing pad at certain points in a call, require that five separate signals be sent from the RTA trunk to the local office. A custom-made IC provided in each trunk provides these "multiple-wink" signals. The ring-forward signal to the toll office is also provided within each RTA trunk. The custom integrated circuit uses a 3-bit binary input set as a command to generate 1 to 5 pulses, each of which turns on (operates) a pair of mercury relays for a timed interval. This pair of mercury relays has contacts arranged in the tip-and-ring leads to provide a reversal to the

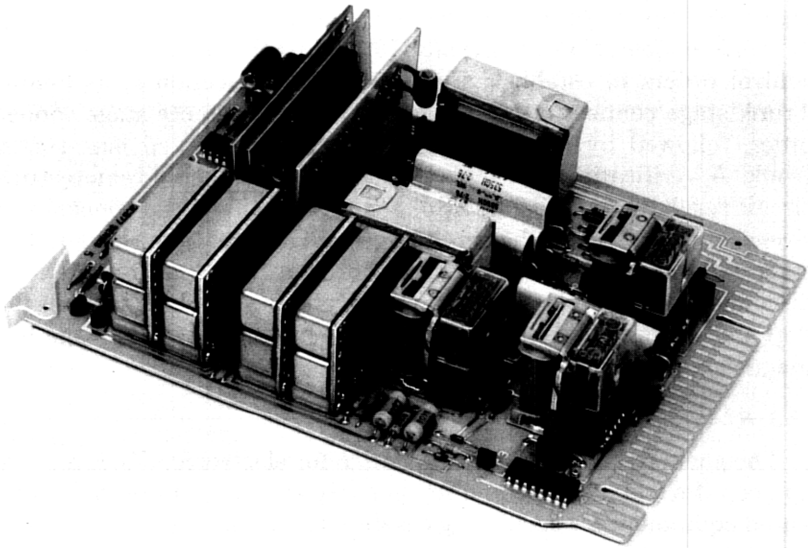

Fig. 7—Typical RTA trunk.

local office. Thus, 1 to 5 timed reversals (winks) can be sent from an RTA trunk. Since this uses only six of the eight possible input states (OFF and one to five winks), the other two are used to operate other relays including a set of "ring-forward" mercury relays which apply +130 V to the toll office side of the trunk. The 3-bit binary input comes from the trunk buffer which generates the 3-bit output from a single SPC order.

With the exception of the use of miniature components and the custom integrated circuit mentioned above, many other talking path and relay states in RTA trunks are similar to previous electronic office trunk designs. The 4-wire trunks utilize the TSPS 3-way 4-wire bridging repeaters mounted on separate repeater bays together with the PWB on the RTA trunk frame. A group of 16 trunks, together with a trunk buffer and appropriate portions of the concentrator controller (Section 2.2) and of the scanner (Section 2.1.2) comprise a *trunk unit.* This trunk unit has each component part arranged to control only the 16 trunks of that unit and is thus completely separate from any other trunk unit. A malfunction in any portion of the trunk unit will thus affect, at most, 16 trunks.

The trunk buffer is similar to all the buffers to which the signal distributor distributes SPC orders and consists essentially of a flip-flop matrix and the necessary control circuitry to set and reset each flip-flop. Output drivers connected to each flip-flop then operate relays in the trunks or send signals to the multiple-wink integrated circuit in each trunk of this particular trunk unit. The trunk buffer is provided with a "super-initialization" input so that, under severe maintenance conditions, an entire buffer or all buffers in the office can be cleared by a single pair of SPC orders. This initialization path is arranged so that all trunks are cleared *except* those trunks in the cut-through (talking) state. Such trunks are *assumed* to be cleared by the initialization routines but are protected by a circuit design that inhibits the action of the clear signal when in the cut-through state. The corresponding program actions are discussed in Section 3.3. This approach allows rapid system initialization when necessary, while at the same time protecting all customer-to-customer talking paths.

2.4 100C position

As mentioned at the start of this chapter, the Position Subsystem (PSS) No. 2 is only briefly mentioned. By adding 100C consoles to the PCL described in Section 2.1, together with some voice path switching and supervisory circuits, a PSS No. 2 is obtained. The 100C position is treated by the PCL signal distributor as one of several "buffer" circuits to which it distributes orders. The single exception to this is the *sequence* of orders required to light up a numerical display (1 to 12 digits) on the 100C console (see Section 2.1.3). A console consists of two positions, and throughout the TSPS literature "100C console" and "100C positions" are used interchangeably.

The upper portion of the position consists of two large keyshelf printed wiring boards and the LED numeric display PWB. Each keyshelf board contains 30 to 50 key/lamps and some integrated circuits and discrete components associated with those lamps and keys. The single "make" contact of each key connects to a cluster of three diodes, which are in turn connected to three leads of a 9-lead bus. Thus each key generates a 3-out-of-9 code on this 9-lead bus whenever it is depressed. The 9-bit bus is examined by the scanner at regular intervals as discussed in Section 2.1.2. The numerical display PWB contains twelve 7-segment LED numerics and the memory and driving electronics associated with them.

The lower portion of the 100C position contains a dc-to-dc converter and several circuit packs of flip-flops, lamp order buffers, and control circuits. The control circuit decodes the serial bitstream from the signal distributor and determines whether it is part of a numerical display sequence or a single lamp order. The numerical display orders are delivered with little change to the LED display PWB. The lamp orders are formed into a vertical select, horizontal select, and group select output set to address a flip-flop matrix. The result is the setting or resetting of a single flip-flop and generation of a check-back signal to inform the signal distributor that the correct conditions for accessing a flip-flop were present (a sort of all-seems-well signal). In many cases, the flip-flop connects directly to a console lamp and simply lights or extinguishes that lamp. However, in some cases a console lamp may be placed in a flashing state and in such cases the flip-flop output connects to a lamp order buffer (LOB). These LOBs generally have two flip-flop inputs and a single lamp driver output. One input is used for a steady lamp "on" condition, while the other is used to connect the lamp to a 60 or 120 IPM interrupter circuit. On a typical call, the TSPS processor might initially send several orders to a position causing three or four lamps to light up and then light and extinguish two or three more in the course of the operator's involvement on the call. For coin calls, part of the numerical display will also be lighted, showing the charge and initial period interval for each call.

In summary, the console contains all the lamps, lamp drivers, memory cells (flip-flops), and overall controls to turn on and hold on the various combinations of lamps required for each call. It contains the circuitry to encode up to 84 keys in a 3-out-of-9 format for the scanner. It also contains duplicated input (signal distributor) and output (scanner) capability and the ability to permit input orders to generate output codes directly, with no operator present, for diagnostic purposes.

2.5 Voice circuits

Although the PCL data lines are voice-grade circuits, they are used to transmit only data between the base and remote locations. Other voice-grade circuits provide connections between the RTA trunks and the base office (base-remote trunks), provide for operator talk paths (for PSS No. 2), and also connect the remotely located maintenance teletypewriter to the base location. Standard 4-wire voice-grade circuits are needed for base-remote trunks, and any type toll-grade carrier system with groups of either 12 or 24 circuits may be used.

For reliability, maximum diversification in the carrier groups, including a spare group, is required. This means that the carrier groups should be evenly distributed over two or more carrier route facilities and, within a route, the carrier groups should be distributed as much as possible over independent terminal equipments and power supplies.

Carrier transfer circuitry is provided at the base and remote locations to switch the spare group into service in place of any other group that fails. Switching is controlled by a program at the base location which requires carrier group alarm signals as an input. Since there is only one spare group, loss of more than one carrier group results in partial facility loss and possible delays in handling calls from that RTA.

III. RTA OPERATIONAL SOFTWARE IMPLEMENTATION

When it was first proposed, the RTA operational software was visualized as being relatively straightforward. At the very beginning of an RTA call, a pair of base-remote trunks would be seized and used to "extend" the remote trunk to the base network. The appearances of the base-remote trunk would be arranged to be the software equivalent of those of a base universal trunk. Thus, throughout the succeeding stages of the call, the existing TSPS operational software[6] could be used without significant modification until, when the operator was released, the base-remote trunks would be disconnected and idled.

Early in the planning, however, it was realized that this was not feasible. Not only was it uneconomical to leave both base-remote trunks up throughout the early stages of a call, but economics also dictated that trunk supervisory states be sensed and controlled via the PCL rather than by sensing and controlling the state of base-remote trunks. Moreover, the technology and complexity of the RTA trunks (see Section 2.3) caused the sequence of orders required to control them to be wholly different from that of base trunks so that the software which constructs trunk orders for base trunks could not be used. Also, the structure of the TSPS operational software did not allow the type of changes required for RTA call processing to be made simply or in a small number of low-level modules.

The result was that the RTA operational software design was forced to evolve away from the initial concept of a few relatively large new modules with most of the existing operational software left intact. Instead, many small changes had to be made to most operational programs in the system. These changes were inserted at points where a position or service circuit is connected or released and at points where trunk control orders are formulated. Typically, the changes involved the addition of a software switch based on a test of the location (base or RTA) of the trunk under consideration. This test is facilitated by bits at several points in the trunk parameter register which have a value of 1 if the trunk is an RTA trunk and 0 otherwise. If the trunk is at the base, the switch directs execution of the previously existing program code. If not, a new RTA program leg is used to handle the call.

Thus, most RTA operational software is relatively straightforward and will not be discussed further. However, several program areas are new to RTA and have no analog in the previous TSPS software. These include the control of the RTA concentrator, the base-remote trunk queuing structures, the "virtual" scanner used to maintain the state of remote trunk supervisory scan points, and a new type of audit. Each of these is discussed in the following sections.

3.1 Concentrator control

During a normal TSPS call, various connections must be made to positions and service circuits. If the call is on an RTA trunk, such a connection requires the selection of:

(*i*) An idle base-remote trunk and an idle path from the RTA trunk through the concentrator to that base-remote trunk and
(*ii*) An idle position or service circuit and an idle path through the TSPS base network from the base-remote trunk to that position or service circuit.

Ideally, the path selection routine should hunt both (*i*) and (*ii*) simultaneously in one unified procedure. Such a procedure would be very complex because it would have to consider the RTA concentrator, the set of base-remote trunks, and the base network as one large, complex, six-stage network (two stages at the RTA and four at the base). However, since the probability of success on (*ii*) is virtually independent of the choice made on (*i*), little effectiveness is lost by hunting the path sequentially: first selecting (*i*) and then (*ii*). This allows one potentially complex single procedure to be replaced by two simpler sequential ones. The first of these is handled by the Concentrator Control program—a new program for RTA—while the second is the previous Network Control program with slight modifications.

Also used with slight modification is the path memory annex register,[6] which is linked to the call register during a network connection. It is used to store the identity of the connected circuits, and the paths used, their states, and, in the case of RTA, the concentrator paths and base-remote trunks used and their states.

The Concentrator Control program is a subroutine capable of hunting, connecting, and breaking paths from RTA trunks through the concentrator to the base via base-remote trunks. It has an interface with the other call processing programs which is similar to that of the Network Control program. Like the latter, it has a set of linkage maps, shown in Fig. 8, which are used in the calculations.

Conceptually, hunting and selecting a path for a call through the RTA concentrator, like any connecting network, may be done in two steps:

(*i*) Find all idle paths from the concentrator inlet of a specified RTA trunk appearance to the base network. If no idle paths exist, the call is said to be "blocked." By "idle path" is meant a combination of an idle base-remote trunk and an idle link connecting the trunk stage switch containing the inlet to the base-remote stage switch containing the base-remote trunk.

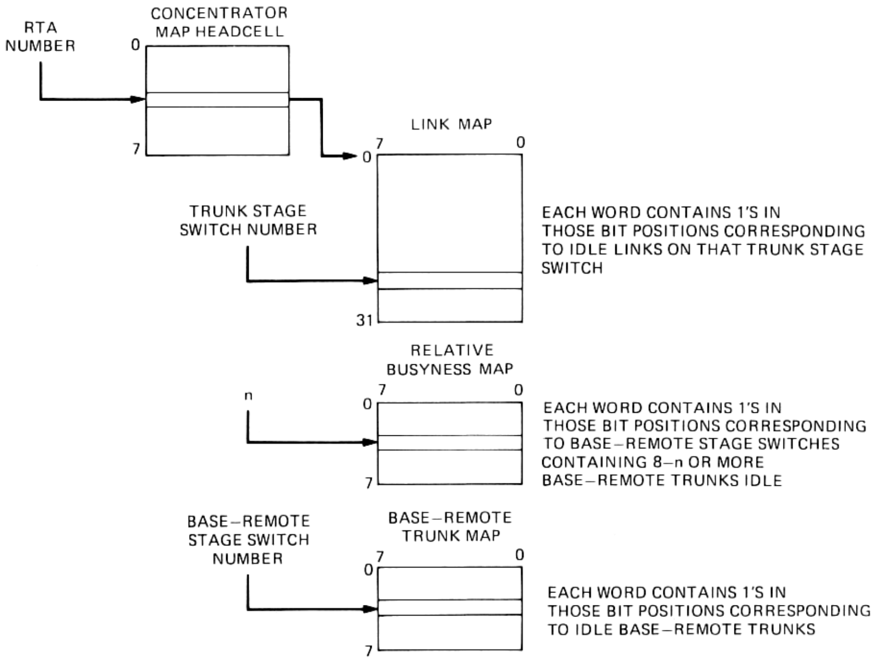

Fig. 8—Concentrator maps.

(*ii*) Select one of these idle paths to be used for the call. The procedure used to do this is called the "routing procedure."

The choice of a routing procedure is important because it affects the blocking probability of the network—the probability that the call is blocked at step (*i*) above— and hence the amount of traffic that the network can carry. One requirement of a routing procedure is that it equalize the long-term mechanical wear of the network crosspoints. Thus, a good routing procedure will minimize network blockage while equalizing crosspoint wear.

An unusual feature of connecting networks such as the RTA concentrator is that none of the routing procedures commonly used for other networks, such as random selection (which is presently used in the TSPS base network) or packing (which is used in the No. 5 and other Crossbar Systems), is a particularly good choice. However, it may be shown[7] that a near-optimal routing procedure is to select the path which goes through the base-remote stage switch which contains the largest number of idle base-remote trunks (with ties broken at random). The base-remote trunk to be used should also be selected at random. Studies show that use of this procedure increases the traffic capacity of the group of base-remote trunks by up to 3/4 Erlang at the engineered load (about 40 Erlangs for a typical, large RTA) compared to the random routing procedure. This means that one less base-remote trunk is typically required—a substantial saving.

The actual implementation of the path hunting procedure integrates the two steps above into one algorithm. First, a test is made to see if all base-remote trunks are busy. If they are, the call is placed on one of the base-remote trunk queues as described in the next section. If one or more base-remote trunks are idle, the routine reads the proper link busy/idle word from the link map (Fig. 8) and forms the logical product of it with word n of the relative busyness map for n as small as possible such that the result is nonzero. A binary search is used to quickly find the proper value of n. If no such n can be found, the call is blocked. If the call is not blocked, then one of the l's in the result is selected at random, the corresponding word of the base-remote trunk map is read, and one of the l's in it is selected at random. The two l's which are selected determine the path to be used by the call. To connect the path, the program loads the appropriate concentrator control orders in an output buffer, updates the concentrator maps, writes the path information into the path memory annex, and then terminates.

3.2 Base-remote trunk queuing

With RTA, a new group of hardware resources, namely base-remote trunks, is added to the TSPS system. One of these resources is routinely applied to a call at the same time as other resources such as operator positions and service circuits. In addition, on some calls, a connection must be made to two base-remote trunks at the same time (on dial-0, dial pulse, non-ANI calls, for example). And most calls have two base-remote trunks up simultaneously at some point in the call (for position and outpulser). Although the normal engineering of these relatively expensive resources is adequate, the supply may be cut in half by an outage of one of the two diverse routes which the base-remote trunks traverse. Since the base-remote trunks are normally engineered near or over 50 percent occupancy, such an outage can instantly throw the RTA into heavy overload. Although rare, this condition is the most likely RTA failure state because of the excellent reliability of the rest of the equipment.

These facts imply that the group of base-remote trunks must be carefully administered. Also, it was desirable to cleanly integrate the queuing for base-remote trunks into the existing TSPS queuing structures which, although they are basically quite powerful and flexible, have the built in assumption that only one resource may be queued for at a time.

The solution chosen is to have two first-come, first-served base-remote trunk queues served in priority order for each RTA.

If all in-service base-remote trunks are busy, calls wait in one of these queues. When a base-remote trunk becomes idle and available for use, it is connected to the first call on the high priority queue. If there are no calls waiting on the high priority queue, the base-remote trunk is connected to the first call on the low priority queue.

Generally, if no base-remote trunks are available the first time in a call when one is needed, the call waits on the low priority queue. Thereafter on that call, the high priority queue is used whenever base-remote trunk congestion is encountered.

This structure has several advantages. First, it gives priority to calls already in the system on which considerable system resources and real time have already been invested. Also it fits well into the existing "one resource per queue" TSPS queuing structure. For example, if a call needs two base-remote trunks at the same time and none are available, the call waits on the low priority queue until it gets one. Then it waits on the high priority queue and it will get the very next base-remote trunk that becomes available (the fact that it received one base-remote trunk while on the low priority queue means the high priority queue was empty). Thus, the call gets two base-remote trunks in slightly more than one waiting time.

Under conditions of RTA overload (caused by, say, a traffic overload or a route outage affecting up to half the base-remote trunks), simulation studies show that the low priority queue can become quite long while the high priority queue remains very short, usually with either zero or one call in it. In this case, every RTA call waits in the low priority queue exactly once, and thereafter uses the very short high priority queue. The result is that every call receives a roughly equal grade of service no matter how many base-remote trunks it needs during its course. (It is impossible for the high priority queue to become very long in the steady state, for if it were to become long, then the low priority queue would never be served and no new RTA calls would ever get into the system—a contradiction.)

The same simulation studies show that, with this queuing structure, the probability of deadlock is *extremely* low. An example of deadlock is as follows. Consider an idle RTA with N in-service base-remote trunks. Suppose that exactly N dial-0 calls arrive nearly simultaneously. Each would be connected to an operator using a base-remote trunk, thus making all such trunks busy. Each operator would then key in the called number and attempt to initiate outpulsing. Since outpulsing requires a second base-remote trunk, all N calls would be loaded onto the high priority queue. Since no base-remote trunks are available, all N calls will be deadlocked until one of the operators or customers abandon. Because the probability of this occurring was determined to be very low, no defensive measures have been added to the programs.

The programs seek and obtain a base-remote trunk for a call, queuing if necessary, before attempting to obtain a position or service circuit. Since positions are a more expensive resource, the base-remote trunk normally remains connected to the call throughout any position queuing to avoid adding the time required to queue for and connect the base-remote trunk to the operator work time. Thus the position queuing time is added to the base-remote trunk holding time. Because this has the potential to translate an operator overload into a base-remote trunk overload, a complex overload procedure is utilized which disconnects the base-remote trunk from the call if the position queue is very long and uses the high priority queue if necessary to reobtain one quickly.

3.3 The virtual scanner

The TSPS programs often determine the supervisory state of a trunk by directly scanning it. This cannot be done with RTA because the RTA scanner is autonomous. To solve this problem, a software image of the state of each trunk scan point is maintained in temporary memory. This is called the virtual scanner, since the required directed scans can be done by "scanning" (a memory read) this area of memory.

It is implemented by adding a virtual scanner (VS) bit to the two bits already assigned to each scan point[6] (one of these gives the last reported supervisory state while the other indicates that reporting supervisory changes has been temporarily suspended on this trunk).

The VS bit represents the software's best estimate of the current state of the scan point and is always updated whenever a trunk supervisory state change report is received from the RTA.

The only complication occurs when an RTA initialization is required. Such an initialization leaves talking state RTA trunks intact, but idles all others. Often when this happens, the information in the virtual scanner is probably unreliable so that the state of the trunk is no longer considered known.

When this occurs, charging is canceled on all RTA talking calls at the time and the RTA trunks are put into a special software "limbo" state. This state means that the software does not know whether the trunk is idle or talking. When an on-hook report is received on a limbo state trunk, the software assumes the trunk was in the talking state and initiates disconnect actions. Conversely, if an off-hook report is received, the trunk is assumed to have been idle and initial seizure actions commence.

3.4 Audits

The existence of the virtual scanner required a new type of audit program called the "virtual scanner update program" which is quite different from other system audits.

Normally, system audits are run on a routinely scheduled basis. Their purpose is to check the integrity of the software data structures. The virtual scanner update, however, is designed to check the consistency of the virtual scanner with the hardware state of the trunks. It does this by sending out a sequence of special orders which cause the trunk to send back several reports which together indicate its current state. Since doing this on all RTA trunks could overload the PCL, it is done very slowly over several minutes. When it is complete, an output message is printed, stating the number of errors found.

The audit is normally run during off-peak night-time hours and is used to verify the integrity of the virtual scanner and resolve any limbo state trunks which may exist at that time. It is also run after certain rare types of fault recognition activity which cause the scanner word buffer to be initialized with the possibility that one or more RTA trunk supervisory reports may be lost.

IV. RTA MAINTENANCE PLAN

Reliability requirements for an RTA subsystem have been set to meet the overall objectives of a TSPS No. 1 system: less than 2 hours downtime per 40 years service and less than one mishandled call per 10,000 calls. The maintenance plan for RTA is further influenced by the fact that an office is frequently unattended and may be located hundreds of miles from the base TSPS.

Sections 4.1 and 4.2 discuss PCL error detection as implemented in the hardware and software, respectively. Remote diagnostics are covered in Section 4.3. Section 4.4 deals with initialization of RTA circuits, while Section 4.5 describes trunk test capabilities for RTA.

4.1 PCL error detection (hardware)

The PCL data lines may be subjected to errors from a variety of causes. To aid in detecting multibit errors which result in single word errors, each word transmitted is encoded with a 5-bit cyclic error detecting code and a parity bit. In addition to single word errors, multiword errors on a data line may occur as a result of hits and fades. Each word error within a multiword error is treated separately by the group gate and remote data circuits. As a result, single and multiword errors are handled similarly.

As mentioned previously, the group gate and the remote data circuit send data to each other over a data line. An error detected by the receiving circuit must be reported back to the sending circuit so that the word can be transmitted again. This is accomplished by a verification system whereby a "scan-complete" is used to acknowledge a report received by the group gate, and a "check-back" acknowledges a properly executed order received by the remote data circuit. If an acknowledgment is not received, the word is retried until it succeeds or the retry limit is reached.

A system requirement of RTA is that the voice circuits and data lines may use any transmission facility. This means that the PCL data lines must be able to operate over voice-grade circuits that have a relatively slow data transmission rate. Because of this and the distance (several hundred miles) between base and remote locations, about 45 ms are required to verify that an order or report has been sent without error. Since a PCL data rate of one order every 25 ms is required to deal with momentary call processing peaks and to allow use of as many existing TSPS No. 1 programs as possible, an overlap operation is employed. This means that, before verification of the first word is received, the second word is sent. If an error is detected, the receiving circuit discards both it and the succeeding word. The sending circuit retransmits both words if it doesn't receive verification of the first word. Therefore, the group gate and remote data circuit are designed to retain information about the last two words sent in case repeat transmissions in either direction are necessary. The overall maintenance strategy is strongly influenced by this aspect of the PCL.

The PCL hardware can detect errors other than transmission failures that would be seen by parity or cyclic code check failures. These faults are classed as "send" or "receive" direction faults. The former type will be discussed first.

The normal mode of PCL operation requires that each PCL half independently handle a processor order until it has been received at the remote location by the remote data circuit for each half. The two remote data circuits then compare the orders and if they match, one PCL half (a so-called "active" half) actually distributes the order to a trunk, concentrator, etc. If one half has a "good" order and the other half has received nothing or has a transmission failure, then the half receiving the good order becomes the active half and distributes the order. This mechanism, known as Full ORed Mode, is intended to reduce order retries due to data line transmission errors.

Each time the processor sends an order via the PCL to the remote location, the buffer which is to handle the order generates a checkback report as verification. This checkback is returned to the processor as an indication of correct operation; its absence indicates unsuccessful operation. The use of checkbacks allows for detection of faults beyond the signal distributor but only partly through the RTA buffers; the actual operation of relays and crosspoints are not verified by checkbacks. Those failures which occur beyond the checkback origination point are detected by error analysis programs.

Due to the overlap operation of the PCL described above, the remote data circuit is designed to also fail the order arriving after an order which results in a checkback failure. This ensures that the second order is not executed before the first, necessary because the second order may depend upon the success of the first order and might lead to a mishandled call if the first order is unsuccessful.

The receive direction faults are those pertaining to the reception of signals originated by RTA circuits. The detection of these faults depends primarily on mismatches between PCL halves. Normally, the two PCL scanners each detect a change of state of an RTA circuit point and generate a report. These two reports are passed back to the group gates where they are compared. Subsequent to receiving the reports, the group gates transmit "scan-complete" words to the remote data circuits so that another pair of reports can be sent to the base.

In addition to matching, however, a cyclic code check is made, an all-seems-well indication from each scanner is checked, a parity check is made, and a sequence code check is made in each report received. The cyclic codes are inserted into the report by the remote data circuit; the all-seems-well, parity, and sequence indications are generated in the scanner. Cyclic codes and parity have been described earlier. A sequence code occupies two bits in each report and is used to eliminate redundant reports received at the group gate. This typically occurs when a "good" report is received but the scan-complete signal fails, causing an unnecessary retransmission of the report by the remote data circuit. If the sequence code check fails, the report is discarded by the group gate. The all-seems-well signal is generated by self-checking circuits within each scanner, as described in Section 2.1.2.

With both PCL halves in service (duplex), one group gate is designated as active. If a mismatch occurs and the active half has a "good" report (one which has correct cyclic code, parity, sequence, and all-seems-well), while the other half has received nothing or has a transmission failure, the good report will be passed on to the processor. However, if the active half has received nothing or has a transmission failure while the other half has a good report, a null scan-complete is generated, causing the report to be retransmitted. This mechanism, known as Half ORed Mode, takes into account the fact that the active group gate controls the matching. (If the program determines that only the active half has a transmission failure or has received nothing, it simply changes the designation of the active half to the one receiving the good report, leaving the PCL in duplex mode.)

In accordance with the circumstances described above, a group gate mismatch results in automatic retransmission of reports. However, if the group gates continuously mismtach several times, the PCL fault recognition program will take action to resolve the mismatch. The tolerance for a few mismatches allows the PCL to automatically recover from a momentary transmission impairment affecting one of the data lines.

Many faults can occur in the scanners which can only be detected by matching. The key characteristic of these faults is the presence of two apparently valid but different words that repeatedly mismatch. The cause of this is either the generation of an extra word or the lack of a generated word in one of the scanners. Under these circumstances, the list of reports in the two memories of the scanners become skewed and a mismatch occurs. The mismatching words represent two scanner inputs, one of which is faulty. This situation is handled by the "receive chain" fault recognition programs, which are discussed in the next section.

4.2 PCL fault recognition (software)

As described in the previous section, the PCL circuits have been designed to detect and correct for most errors which are the result of transmission impairments of short duration. For longer hits, fades, and PCL circuit failures, various other hardware indications, such as check-back failures and group gate alarm ferrods, allow PCL fault recognition programs to maintain working send and receive chains. The send chain refers to those circuits involved in the distribution of a PCL order: the group gate, remote data circuit, signal distributor, and the PCL buffers. The receive chain refers to those circuits involved in the reception of a PCL report: the group gate, remote data circuit, and position and trunk scanner.

If an order has been retried on a duplex PCL without the reception of two successive check-backs for 0.5 second, the send chain fault recognition program will be entered. When this occurs, call processing is suspended and a special test order is sent to an independent buffer. If that order is executed successfully, the original buffer accessed is deemed faulty. If the test order fails, the program splits the PCL so that each PCL half is completely independent with no matching taking place. The resulting simplex operation allows the original failing order to be tried first on one and then on the other PCL half. If the order fails over both halves, the program tries a second test order (to a different buffer) over each half. Based on the results of these tests, the program decides which half is faulty and places that PCL half out of service. Normal call processing is resumed on the good half, and diagnostics are run on the out-of-service half.

For receive chain faults, the fault recognition programs are entered whenever an error rate monitor (administered by software) corresponding to a group gate alarm ferrod exceeds its threshold. The absence of either all-seems-well, good parity, or proper sequence on a PCL half automatically indicates the faulty half. Mismatches between two valid reports involve more detailed processing.

Due to the overlap operation of the PCL, two reports on each half are sent to the base during retransmission. In the case of skewed reports, chances are that two of the reports received match; that is, one of the reports received on one half may match one of the reports on the other half. The receive chain program checks for this case to eliminate the need for testing all four reports. Only those reports seen by only one half are further tested.

When the receive chain fault recognition program is entered, call processing is suspended. The PCL is again split into two simplex paths and a bypass in the scanners is activated to allow test results to be sent back to the base without disturbing the scanner word buffers. Each scanner is then checked, via test points, for the ability to generate the reports to be tested. These test results are analyzed by the program to determine which PCL half is in trouble. The faulty half is placed out of service, and diagnostics are run on it. Call processing resumes on the in-service half. The flow of reports proceeds from where it was interrupted, with the PCL in the simplex mode. While in the simplex mode, subsequent faults of this type go undetected.

In general, fault recognition program actions are transparent to call processing programs. In the case of a PCL mismatch, the original reports are retained in processor memory so that, once it is determined which PCL half is faulty, the reports already received on the good half can be passed to call processing programs. As already mentioned, the PCL scanners contain word buffers that can retain up to 31 reports which may have occurred while fault recognition tests were being applied. This is sufficient memory to handle a back-up of reports caused by fault recognition activity in all but the very worst cases.

In addition to the send chain and receive chain fault recognition programs, the data line fault recognition program responds to transmission errors reported to the SPC. Error rate monitors, driven by the group gate alarm ferrods, are maintained for each connected data line. When the allowable error rate is exceeded, the line that appears to be faulty is replaced by the spare line. After a delay of a few minutes, the apparently faulty line is reconnected. If its error rate is acceptable, no further action is necessary. If it fails again, however, the spare line is used until the faulty line is repaired or has been determined by subsequent automatic tests to have recovered. The spare data line is also used periodically so that its operational status can be monitored.

The state of the data lines must also be considered by the fault recognition programs when reconfiguration is being performed. If all circuits on a PCL half are working but both data lines that can connect to it are out of service, then that PCL half is not usable.

Although the normal mode of the PCL is the duplex mode, it is possible that a fault could occur while the PCL is simplex. The PCL could be simplex because diagnostics are being run on the out-of-service half or because a circuit on that half is inoperable. If the fault recognition program determines that the active half has a circuit in trouble while the PCL is in simplex, the out-of-service half will be placed in service and the active half removed from service.

In cases where the out-of-service PCL half is already marked in trouble or has been manually made unavailable to fault recognition and a fault has also been detected on the in-service half, no known working path of PCL units exists between the base and remote locations. Rather than to place the entire PCL out-of-service, a special fault recognition program is invoked which attempts to find a working path, regardless of the software status of the units. This program systematically tests both the send chain and the receive chain of the PCL by using test orders and different combinations of data lines and base peripheral circuits that interface the processor to the group gates. In the event that these tests fail, the PCL will then be placed out of service. However, the program will automatically retest the PCL every few minutes and restore it to service as soon as one working (simplex) path is found. Because the duplex state offers maximum error detection, an out-of-service PCL half is also automatically tested for the absence of units in trouble (or a manual request to leave the half out of service) when the PCL is simplex. If no reason for the simplex configuration is found, the PCL will be restored to the duplex mode.

4.3 Remote diagnostics

The long data paths to RTA and PSS No. 2 installations and the desire to use voiceband facilities resulted in the use of 2400 b/s data rates between the SPC location and the remote locations. If lengthy strings of diagnostic tests were run from the SPC over an in-service PCL half to locate a trouble in the other (bad) PCL half, then call processing would be virtually halted for the duration of the diagnostic. To prevent such adverse call processing effects and still allow thorough diagnostics, the entire set of diagnostic test sequences are placed at the remote end and controlled by a diagnostic control circuit (DCC) which, in turn, is controlled by the SPC. The concept of placing the diagnostics *at* the remote location provides two other very desirable operations as well as minimizing adverse call processing effects. It allows the tests to run very fast since no transmission time is involved, and it allows the DCC to have direct access to various internal points of the circuits being diagnosed which would otherwise be impossible to obtain. With the exception of the scanner with its need for 25 ms to recognize a change of state (see Section 2.1.2), diagnostic orders can run at least an order of magnitude faster than regular PCL orders. This significantly reduces the time between a request for a diagnostic and TTY output of results.

The DCC is essentially a very low-level minicomputer with sequences of tests and expected results of those tests stored in Read-Only-Memory (ROM). Each ROM word is 24 bits long and contains a 3-bit operation code (op code), 20 data bits, and a parity bit. The data bits usually apply a set of test conditions to a PCL circuit and cause that circuit to take some action. A subsequent ROM word contains an expected result which is matched against the actual result. If the result agrees, then another test word is selected from ROM and the process continues. The access that the DCC has to the various PCL circuits is shown in Fig. 9. Note that *indirectly* the DCC has access to *all* remote circuits; e.g., although it has no direct connection to a 100C console it can send orders to the RDC which can cause the signal distributor to send an order to a console. Similarly, while it has no input from a console it can check the scanner output to see if the scanner received a console output.

When comparisons of actual and expected results fail, a reply (noting the failure) is sent back to the SPC over the in-service PCL half. However, if massive failures are encountered, this reply capability is limited, so as to limit its effect on call processing. Also, massive failures will cause the SPC to terminate the diagnostic early since they quickly pinpoint a source of trouble. Whether or not comparison failures occur, every 64th ROM word causes a "progress report" type reply to the SPC to inform it that the diagnostic is continuing. The "failure report" sent back to the SPC on all mismatches provides a number between 0 and 63. The first 64 words of ROM store these 64 possible failure reports. The action taken by the DCC when a comparison failure (expected result versus actual result) occurs is to just set all but the least

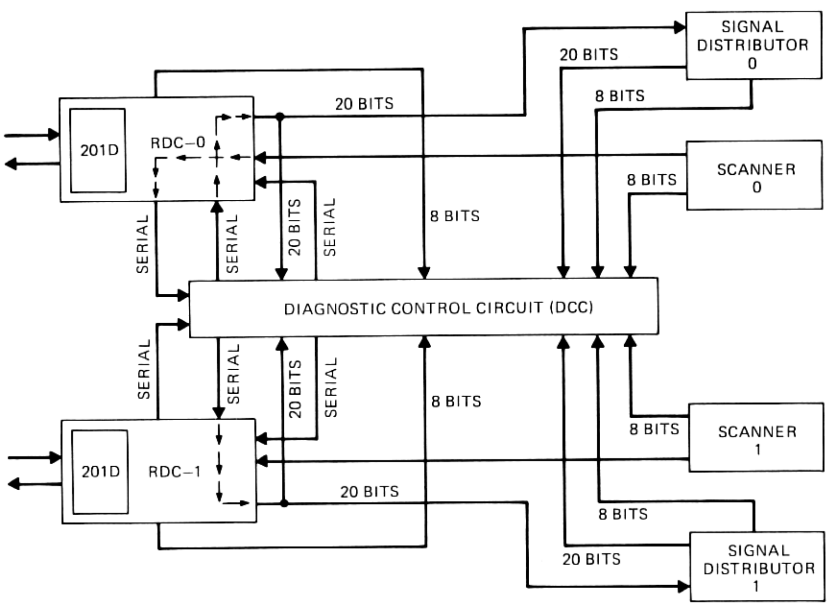

Fig. 9—DCC access to PCL circuits.

significant 6 bits of the address counter to zero. The next ROM word selected will then be one of the first 64 words and as this reply is sent back to the SPC and upper address counter bits are unclamped. With all address counter bits operational, the next word selected is the next word in the diagnostic test sequence. Thus no real program transfer and return function is needed in the DCC since this hardware approach to changing the address counter output does the transfer job and automatically includes the correct return condition.

The SPC controls the DCC by three steps. First, an order is sent over the inservice PCL half to place the out-of-service half RDC in the diagnostic mode, i.e., connect it to the DCC. Second, an order is sent to the DCC (again, over the in-service PCL half) to preload the desired starting address into the DCC address counter. Finally, a "start" command is sent to the DCC to select the first ROM word (at the starting address). This first word of any diagnostic sequence is always an "SPC reply" type word that returns the starting address to the SPC. The SPC checks this against the desired starting address and stops the DCC if the return is incorrect.

Since the SPC knows the starting address and is informed when every 64th subsequent address is reached, it can quickly compute the exact location of a comparison failure (its location in *that* diagnostic sequence) whenever it receives a failure report with its 0-63 tag number. Thus at the end of a diagnostic sequence, the SPC has a pattern of failures and uses these to generate a "trouble number." This number is then compared to a trouble dictionary to locate the failure.

The overall access which the DCC has to each remote RTA circuit can be understood by looking at the 8 possible operation codes together with the expansion of one such operation code. The 8 operation codes (op-codes) are listed below. This listing also gives the reader a glimpse at the DCC programming possibilities and indicates the ease with which each RTA circuit designer could write the appropriate diagnostic.

000 8-bit comparison or 25-ms delay
001 20-bit comparison from RDC
010 RDC test vector (DCC output)
011 20-bit scanner comparison preceded by 25 ms delay
100 Signal distributor test vector (DCC output)
101 20-bit comparison from signal distributor
110 RDC test vector (DCC output)
111 SPC reply word (DCC output)

The 000 op code is expanded within the 20-bit data field since only eight of the data bits are actually needed for an 8-bit comparison. The order has the form:

23	22							15			11				6				2		0
P	Z_7	Z_6	Z_5	Z_4	Z_3	Z_2	Z_1	Z_0	1	1	1	C/D	X_3	X_2	X_1	1	M	C	B	A	000

where

Bits 0 to 2 are the op code (000).

CBA = 3 bits used to select the particular 8-lead input desired.

C	*B*	*A*	*Compare Input*
0	0	0	Input from signal distributor
0	0	1	All ones input
0	1	0	Input from RDC
0	1	1	Input from scanner
1	0	0	Spare input set Y
1	0	1	Spare input set X
1	1	0	Input from DCC
1	1	1	All zero input

M = "Mask" bit used to insert 1-bit mask. If M = 1, no mask is used and all 8 bits specified by CBA are compared against $Z_7 - Z_0$. If M = 0, a 1-bit mask is enabled as specified by the $X_3X_2X_1$ bits.

$X_3X_2X_1$ = Mask selection bits used to specify which of the 8 input leads are to be compared.

C/D = Compare/delay bit used to differentiate between use of this order for 8-bit comparisons or for 25-ms delay intervals.

$Z_7 - Z_0$ = 8 comparison bits.

The other seven op codes perform only the function listed and all 20 data bits are either applied by the DCC as a test vector or used by the DCC for a comparison.

The op code listing together with the 8-bit comparison possibilities shown above indicate the large number of internal points of all remote PCL circuits to which the DCC has access. The DCC concept allowed diagnostic points to be placed where they would do the most good. This makes every test/comparison pair of orders a powerful pair and leads to short diagnostic phases. These short phases together with the fast execution times allow most diagnostics to complete in a few seconds. As the diagnostic completes one phase (and before it starts a subsequent phase), a few SPC orders can be sent out to reconfigure the PCL half under test. For example, the RDC can be configured to loop back on itself. The DCC can then send orders to the scanner input side of the RDC—similar to reports normally returned by the RDC to the SPC—and, since the RDC is looped, these orders will come right back through the receiving half of the RDC. They can then be examined by the DCC or allowed to continue to the Signal Distributor and cause some action there. Again, at the Signal Distributor, the test vector can be examined by the DCC or allowed to propagate even further and cause some other action further downstream. Judicious use of such DCC comparison tests allow very thorough diagnosis of remote circuits.

The DCC also has inputs from itself as well as all 1 and all 0 type inputs. It can use these inputs to run tests on itself and inform the SPC of possible DCC troubles. Massive failures in other circuits usually trigger (via an SPC program decision) this self-test of the DCC to ensure that the massive failures are real and not just the result of an inability of the DCC to make comparisons.

The DCC currently uses about eleven thousand 24-bit ROM words to test out all remote PCL circuits. The memory layout provides three circuit pack locations, each of which can contain $8K$ (K=1024) words so the memory can grow to 24K words if needed. The upper 16K addresses are also available in eight other circuit pack locations, each of which can accommodate a 2K-word PROM circuit pack. With this arrangement, the memory can be placed on ROMs on 1-1/2 circuit packs. Then, as various remote PCL circuit changes and additions take place which would tend to degrade the original diagnostic effectiveness, new versions of selected diagnostic phases can be placed in PROMs on, say, one new memory board plugged into a PROM circuit pack location. A 1-word change in the SPC program can then be used to specify a new starting address for the affected phase (or phases). The old version of the diagnostic is still there, but never gets used while the new (higher numbered) starting address selects *that one phase* from the PROM board. As PROMs are added over a period of years, and several sections of the original (ROM) program become unused, the ROM can then be redone to reflect several years of change.

4.4 RTA initialization

The RTA maintenance buffers contain a variety of distributor points used in conjunction with the operation of several remote circuits. When PCL reconfigurations occur, either by manual request or as a result of fault recognition action, some points in each buffer must be cleared. Some points are used to initialize the remote data circuit, signal distributor, and position and trunk scanner. Each buffer also contains points not connected with a particular PCL half which should not be cleared when reconfiguration takes place. To achieve the desired results, each maintenance buffer has been functionally divided such that selective initializations of these buffers can be performed. The selective initialization feature of the maintenance buffers also provides the capability of regenerating the PCL status display at the remote location upon manual request. In addition to the selective initialization leads, each maintenance buffer is equipped with a lead which clears all points in the buffer. These leads are pulsed only when the entire PCL is being initialized, such as when a system initialization occurs, or when the PCL has developed faults on both halves.

Under certain circumstances (e.g., RTA or TSPS system initialization), it is desirable to place all the remote and base-remote trunks in an idle state. Since the number of remote trunks (496 max) and base-remote trunks (64 max) in an RTA may be large, individual trunk initialization would be very time-consuming. The trunk super-initialization feature of RTA allows all the remote trunks to be initialized with two orders and all the base-remote trunks to be initialized with one order. This allows for speedy initialization of the RTA trunks when necessary. It should be noted that those trunks already in the talking state will not be affected by the super-initialization orders.

4.5 Test frame capabilities

The test frame at the remote location displays status information and provides control and test facilities including a maintenance teletypewriter. As shown in Fig. 10, the out-of-service status lamps for the PCL equipment are physically arranged to depict the equipment ar-

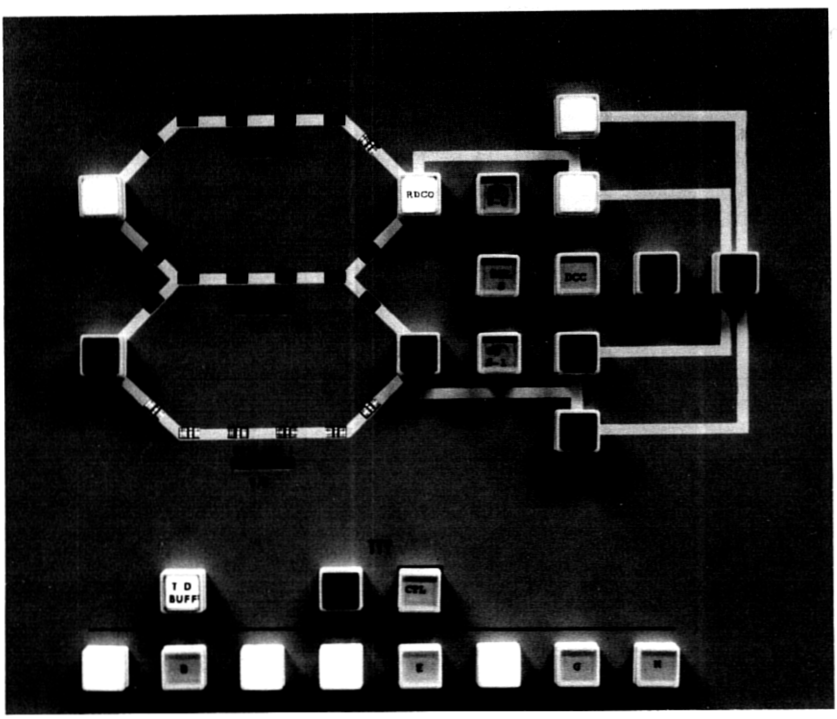

Fig. 10—PCL and voice status displays.

rangement in block diagram form. The three data lines are represented by three rows of light-emitting diodes which, when lit, indicate active working routes. The SW 0, SW REL, and SW 1 keys in the status display enable the remote ends of the data lines to be switched. This capability is provided to allow manual intervention in the rare case that both active data lines fail before the maintenance program at the base location has the opportunity to order the remote equipment to switch.

Several trunk test facilities are provided on the test frame as indicated in Fig. 11. A Dial Access Test Line (DATL) permits local office personnel to check transmission and noise levels on incoming trunks between the local office and the RTA without requiring assistance at the RTA. The Master Test Line (MTL) permits RTA personnel to establish voice communication with any trunk that terminates on the concentrator. The access line permits any trunk to be connected to any one of the test frame facilities which include a voltmeter, transmission level and noise measurement apparatus, a variable frequency milliwatt supply, and a quiet termination.

The test frame voltmeter and transmission measuring devices have a digital readout capability. The measurements can be remotely read

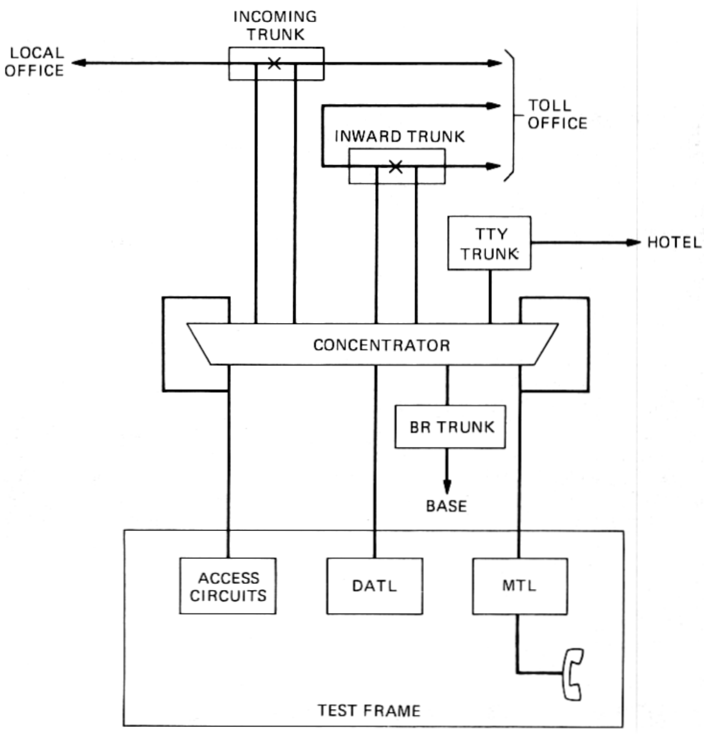

Fig. 11—Trunk test facilities.

by the base program and printed on the base location maintenance teletypewriter. This permits detection, verification, and sectionalization of many trunk problems without anyone present at the RTA.

V. ACKNOWLEDGMENTS

The design and implementation of the RTA hardware and software benefited from the efforts of many persons other than the authors. We would like to particularly acknowledge the contribution of P. S. Bogusz, T. J. Bradley, K. A. Heller, T. Huang, L. C. Kelly, R. V. Kraus, R. N. Markson, J. J. Rielinger, L. A. Rigazio, R. J. Roszkowski, S. M. Silverberg, R. A. Tengelson, R. J. Thornberry, R. A. Weber, and G. Y. Wyatt.

REFERENCES

1. E. Tammaru, "Correlated Error-Rate Results in Data Links with Duplicated Channels," National Telecommunications Conference Record (December 1975), p. 42.26.
2. E. O. Elliot, "Estimates of Error Rates for Codes on Burst-Noise Channels," B.S.T.J., *42*, No. 7 (September 1963), pp. 1977–97.
3. J. A. Hackett, et al, "TSPS/RTA Reliability and Maintainability in the Hardware Design," ICC 75, *3*, pp. 46-5—46-10.
4. A. F. Bulfer and B. D. Wagner, "Calculating Delay and Buffer Overflow Rate of Feedback Data Channels," 7th Hawaii International Conference on System Science, 1974.
5. F. H. Smith and J. M. Catteral, "Uses for the New Crossbar Switch," Bell Laboratories Record, *49*, No. 7 (August 1971).
6. A. W. Kettley, et al, "TSPS No. 1—Operational Programs," B.S.T.J., *49*, No. 10 (December 1970), pp. 2625–2685.
7. A. F. Bulfer, "Blocking and Routing in Two-Stage Concentrators," National Telecommunications Conference Record, December 1975, pp. 19-26 to 19-31.