Measuring Benefits of Controller-Pilot Data-Link Communication (CPDLC) System in an Airport Area Using a Microscopic Simulation Model

(Words count: 7,288 = 4538 + 9 Figures + 2 Tables)

Hojong Baik
Research Scientist
Department of Civil an Environmental Engineering
Virginia Tech
Blacksburg, VA 24061
hbaik@vt.edu
Ph. 540-231-2362
Fax: 540-231-7532

Antonio A. Trani
Associate Professor
Department of Civil an Environmental Engineering
Virginia Tech
Blacksburg, VA 24061
vuela@vt.edu
Ph. 540-231-4418
Fax: 540-231-7532
Abstract:

The Federal Aviation Administration (FAA) plans to implement a controller-pilot data-link communication (CPDLC) system in the near future. Unlike conventional voice-channel communication systems, the CPDLC system allows pilot and controllers to exchange messages via specially designated data links thus reducing voice communication congestion. Under the proposed CPDLC system, pilots can see an air traffic controller message in text format on a Multifunction Display (MFD). Similarly, pilots can send a message to controllers using pre-coded messages via the MFD. By reducing communication times, the CPDLC system is expected to reduce flight delays at congested airports and in the airspace. The system is also expected to decrease operational and communication errors that could lead to fatalities. The CPDLC system can be implemented in multiple domains in the air traffic control system including airport area, Terminal Radar Approach Control (TRACON) areas, Air Route Traffic Control Centers (ARTCC). Previous studies have focused on the evaluation of CPDLC benefits in either the TRACON or ARTCC domains. In this paper, we assess the benefit of implementing CPDLC system in an airport area using a microscopic airport simulation model.
INTRODUCTION

One of the primary goals of the Air Traffic Control (ATC) system is to provide secure, continuous and seamless communication between controllers and pilots. The current ATC system heavily relies on the voice radio communication (‘voice communication’ in short will be used through this paper). The concept is analogous to a conference call model where many attendees talk over a single telephone line. In the voice communication system the controller and all pilots under the jurisdiction of one controller, talk over the same communication frequency. Since the controller can talk only to a pilot at a time, waiting times for the voice channel is a natural phenomena given the randomness of the calls and the limited capacity of one channel frequency. In many instances messages that are not clearly understood by the controller or a pilot, require multiple repetitions thus wasting valuable communication time. If the amount of communication messages approaches the single channel communication capacity the system behaves like a stochastic queueing model and message delays increase rapidly as the capacity of the channel is approached. As number of flights in National Air Space (NAS) is expected to increase over time, the amount of controller-pilot communications is also expected to increase accordingly. Some enroute and terminal area airspace sectors in the National Airspace System (NAS) already experience frequency congestion during peak traffic periods and as a result aircraft are delayed.

Unlike voice communication system that occupies the communication channel continuously, the Controller-Pilot Data Link Communication (CPDLC) system spares the frequency by exchanging the messages via a specially designed electronic data link. The CPDLC messages are then displayed in a text format on a Multifunction Display (MFD) inside the cockpit. An example of the CPDLC message on the MFD box is shown in Figures 1 and 2. Data link messages can be sent by pilots to controllers (or reciprocally) via a standard keyboard interface. CPDLC reduces frequency congestion and could potentially reduce flight delays. Another potential benefit expected from CPDLC is the reduction or elimination of miscommunications between pilots and controllers that could possibly lead to critical mishaps. For these important reasons, the Federal Aviation Administration (FAA) considers the CPDLC system as the future communication system in NAS.

The CPDLC system has been tested by FAA for one year (from October 7, 2002 to October 6, 2003) in the Miami area with a group of specially equipped American Airlines aircraft. The results show that during the test period, over 40,000 messages were exchanged via CPDLC on 3,173 flights, and no operational error and no pilot deviations were observed. It is also reported that 23.8 hours of communication times were saved compared to the voice radio communication. FAA has plans to deploy the CPDLC system in high-altitude enroute control sectors starting in December 2005. [FAA]

Up to now, the CPDLC has been primarily designed and tested in the en route sector or TRACON sector domains. CPDLC has also been used for over a decade in aircraft to controller communications on the ground to transmit clearance delivery functions to the ATC system using ACARS. It should be noted that the benefits of CPDLC could easily be extended to ground controller-pilot communications at busy airports where delays are frequently observed not only due to the lack of infrastructure capacity but also due to the congestion of voice radio communication channel.

Motivated by this fact, this paper focuses on assessing total delays occurring at the airport area under two communication systems: 1) the voice communication system and 2) CPDLC system. The remaining of this paper is organized as follows. In the next section, the differences of two communication systems, i.e., voice radio communication system and CPDLC, are described more formally using so-called ‘event diagrams’ that are commonly used in the simulation modeling domain. In the following section, a microscopic simulation model developed by the authors is briefly introduced as a tool to study aircraft and controllers’ behavior in great detail. Implementation of two communication systems in the simulation model is also described in this section. Using the Ronald Reagan National Airport (DCA) airport and various flight schedules to represent different demand levels, we measure the total operation times and average delays per flight using the two different communication systems. In the final section, we conclude this paper summarizing relevant findings and we provide several recommendations for future study.
PREVIOUS STUDIES

FAA have conducted two ‘man in the loop’ simulation studies to analyze the benefit of applying two-way Data Link communications mixed with voice radio communications. [FAA, 1995, 1996] These studies were designed to duplicate air traffic flows sampled from sectors in the Atlanta ARTCC and in the Newark area of the New York TRACON. In order for the studies to be able to measure the efficiency of CPDLC at different demand levels, experiments were designed to gradually increase the aircraft traffic from the baseline demand taken from the System Analysis Recording (SAR). The measures of efficiency in these studies are the flight distances traversed by individual flights, flight times across sector boundaries, flight delays, etc. The studies concluded that the CPDLC system could reduce the number of voice messages sent by controllers by up to 66 percent, which consequently results in an increase in the arrival rate, reducing flight delays, enhancing safety and reducing controller workload. It is also reported that the extrapolated benefit for the National Airspace System (NAS) would be up to $337 million annually.

Rodgers et al. [1997] evaluated the benefits of CPDLC in the enroute environment for different equipage rates using a discrete event simulation model. The model was designed to portray the transmission of communication messages via two communication systems: voice and CPDLC. This study concluded that with 90 percent of equipage rate, the voice channel occupancy can decrease up to by 91 percent in the Denver ARTCC. Massimini et al. (2000) estimated the voice channel occupancy in a sector by manipulating output from the Total Airspace and Airport Modeler (TAAM). TAAM is a microscopic discrete simulation model. Kern [1991] and Ryan [1992] studied the performance characteristics of a system employing both the voice radio communication and the CPDLC systems together. Ryan suggested that with a combined system that relies 90 percent on CPDLC and 10 percent on voice radio communication system, the FAA and airlines could achieve operational benefits.

Studies on workload for controller or pilots under the CPDLC environment can be found in Groes [1987], and Midkiff et al. [1992], Corwin et al [1990], Reynolds [1990], Hahn et al [1992], and Nelson [1996] addressed the requirements or architecture of CPDLC systems. Several simulation studies that have been conducted by NASA include Hinton et al. [1988], Waller et al [1989], Knox et al. [1991] and Mackintosh [1999]. All previous studies related to the analysis of CPDLC impacts are summarized in Sen [2003]. To our knowledge, there is no study conducted to assess the benefit of employing CPDLC systems in the airport area where CPDLC could also contribute to reducing congestion.

DIFFERENCES IN TWO COMMUNICATION SYSTEMS

In order to discuss the differences of two communication systems, we first need to understand the basic control system employed in the airport area. In general, two types of controllers are cooperating together in the control tower to manage arriving and departing aircraft at an airport: 1) local controllers who are responsible for runway operations i.e., takeoffs and landings, and 2) ground controllers who are responsible for taxiway operations from gates to runways and reciprocally from runways to gates. Controllers use so-called flight progress strips (or flight strips for short) to store critical flight information for each handled flight. This information includes flight number, aircraft type, origin airport, destination airport, arriving route, etc. This information is printed in a rectangular piece of paper, and stored in a plastic holder while the flight is under supervision of an air traffic controller. New ATC hardware systems provide flight progress strips electronically with similar functionality. In order to portray these flight strips, three types of flight strip bins are modeled from the controller’s viewpoint: pending, processing and completed flights bins. This is illustrated in Figure 3.

For modeling convenience, we group the communication instances into two modes depending on who initiates the communication: 1) passive control mode in which the controller renders control messages replying to various flight requests, and 2) active control mode in which the controller makes decisions to avoid possible aircraft conflicts or to control the overall traffic flow. At the start of the simulation, the controller state is “standby” and all flight strips are in the pending box. Receiving a request from a flight in the passive control mode, the controller places the corresponding flight strip in the processing state and begins to judge the situation. The controller’s judgment depends on the flight’s current state and traffic situation. For example, if a flight is at the final stage in the arrival
process and it requests a clearance for landing, the controller needs to check the flight’s relative position to other flights around it, and then makes a decision on whether or not to allow this flight to proceed by checking the minimum separations between flights. The controller then sends a control message to the flight and waits for confirmation. Once a confirmation is received from the flight, the controller state returns to the "standby" status and waits for another request. The state transition diagram for the local controller is depicted in Figure 4.

When a flight finishes its operation completely, the controller moves the flight strip from the processing bin to the completion bin. Whereas, if a flight is still performing its operation but has passed the limits of a controller’s custody (i.e., control boundary), then the current controller hands the flight strip over the next controller's processing bin. At the moment when an arriving flight exits from a runway after completion of the runway landing roll and starts taxiing to gate, for example, the flight strip for the arriving flight moves from the local controller's processing bin to the ground controller's processing bin.

If any traffic congestion or conflict is expected, controllers can intervene during aircraft taxiing, issuing control messages to slow-down, speed-up, or even to stop aircraft at the current position. This active control decision-making process is largely based on the controller’s experience, subject to ATC rules enforced by the FAA. It is difficult to devise a single comprehensive traffic management rule that is applicable to all ground control cases. From the modeling point of view, it might be more practical to develop a rule-based decision making process for the active control process. A controller also initiates an active control process when the controller is in "standby" and there is some flight in the state of "wait for controller's contact". In fact a controller realizes the existence of flights awaiting controller's contact by checking the strips in the processing bin. The communication process initiated by controller is shown in Figure 5.

Regardless of the type of communication system, a communication task is completed with four different steps. From a pilot standpoint, the steps are: 1) sending a request, 2) waiting for a command, 3) receiving a command, and 4) sending a confirmation. The corresponding steps for an air traffic controller are: 1) receiving a request, 2) judging the request, 3) sending a command, and 4) receiving a confirmation. In the model, the time durations required for each step are specified as $t_1$, $t_2$, $t_3$ and $t_4$, respectively. In the model, the time required for each step varies depending on the communication system. Compared to the voice communication, the data link system which is an advanced system using electronic data transmission requires small delay times for either sending requests or receiving messages. However, it is assumed that the times required for making a judgment are same in both systems.

Let us describe this situation in detail. Regardless of the type of communication system, the flight’s communication state is initially set to “readyToCommunicate”. Once the flight needs to communicate with a controller, then it tries to send a request, which will be accepted by the controller unless the controller is busy. If the controller is busy, the flight has to wait until the controller completes the current communication. The first difference between the two communication systems occurs at this point. Under the voice communication system, the flight has to wait for a certain amount of time ($t_0$) and is required to try again to contact the controller. In this case, the flight state changes to "wait for next communication".

In the datalink system, the flight has to wait until the controller completes the current communication, but the flight does not need to send to the controller the request again. In this model it inherently assumed that the datalink system can store and show information on the controller's screen with requests the controller could not handle immediately. In the datalink system the flight needs to wait for the controller contact. In this case, the flight state changes to "wait for contact from controller" If the controller is in a "standby" state when the flight attempts a contact, the controller state is set to "busy". Once a flight succeeds in contacting the controller, it communicates exclusively with the controller until the entire communication phase is completed. After the flight confirms the command, the flight and the controller states are set to “readyToComm” and “standby”, respectively.

If the taxiway or runway system is congested by the traffic, it is possible that the currently contacting flight fails to obtain the “clearance” for its operation. If this occurs, the flight needs to wait until the controller contacts the flight. In this case, the flight communication state is set to "wait for contact from controller ". The state transition diagrams for both communication systems are depicted in Figures 6 and 7.

A Microscopic Airport Simulation Model

To assess the implications of two communication systems in the capacity of the airport we use a microscopic computer simulation model developed at Virginia Tech. The Virginia Tech Airport Simulation Model (VTASIM)
was developed by the authors to study the effects of advanced technologies at airports. The model is classified as a microscopic, hybrid and deterministic simulation model where: 1) activities are traced and recorded individually for every aircraft, 2) it has components of discrete and continuous simulation models, and 3) it does not explain entities stochastic behaviors which requires extensive statistical analyses. For more details, see Baik and Trani [2004]. The main part of the simulation model is executed in such a way that the states of all entities i.e., flights, ground and local controllers in the system are updated every time interval (typically every second) until the simulation time ends. The state variables for entities and initial values are summarized in Table 1.

Figure 8, two types of computational loops involved in the execution of the model: an outer loop where the system clock proceeds by a time increment ($\Delta t$) until the simulation ends, and an inner loop where the states of all flights in the system are checked and updated successively. Inside the inner loop, a check is made for each flight clock time to determine the flight movement. The flight clock time is initially set to the time given in the flight schedule. When the system clock advances and passes the flight scheduled time, the flight executes two major activities related to communication and movement within the outer loop. Otherwise, all the processes inside the outer loop are just skipped.

Unlike the flight movement, which is checked continuously, the communication activities are treated as discrete events. This is because a communication event is scheduled only when it is necessary. Communication states in both controller and flight objects change in a discrete fashion. The module named “checkNeedToComm” checks if a flight requires communication with either the local or the ground controller. If a flight attempts to communicate, two state variables, “needToComm” and “nextCommEventTime” are set to “true” and the current system time, respectively. The communication module initiates the communication events and changes the flight movement state if it is permitted to move.

The states of the flight movement such as speed, acceleration, position, etc., are continuously evaluated after a flight enters the system. The main concern in the movement logic is to decide how much the flight would accelerate (or decelerate) in the next time interval. Unless a flight is either stopped or parked at a gate, or waiting on a runway, the flight dynamic behavior is decided by its own control logic depending on its current movement state. For example, if an arriving flight is in the coasting phase on the runway, its acceleration for the next time interval is determined by a second-order feedback control system. The acceleration (or deceleration) logic during taxiing is a little more complicated because it might depend on a leading aircraft, on potential collisions with others at intersections, and on the remaining distance to the destination for taxiing. The model is designed to be able to decide the acceleration (or deceleration) for a taxiing aircraft based on a conflict detection-resolution algorithm imbedded in the model. The last procedure inside the inner loop is to update the flight kinematic states to reflect the current changes according to the new acceleration value selected. In the case that a flight enters a new link, the flight information on its leading and following flight as well as the link information are updated at this moment.

Once the complete procedure for a flight has been executed, a check is made to decide if this flight is the last one on the active flight list. If it is the last flight, the inner loop is completed, and both ground and local controllers start to check if there is any flight awaiting contact by looking at their flight progress strips. If the controller finds any flight waiting and the controller is in “standby” state, contact is made by the controller with that flight. Once all controllers finish appropriate actions, the simulation time advances by a time increment to commence another iteration of the outer loop.

**CASE STUDY**

To understand the impacts of CPDLC technologies in the capacity of an airport, we simulate operations at one hub airport with various demand levels. The following paragraphs describe the scenarios employed in this study.
1. Input Data

A. Airport

The Ronald Reagan National Airport (DCA) is selected as a case study airport for the CPDLC analysis. The existing DCA Airport has 45 gates and three crossing runways designated 3/21 (4,506 ft), 15/33 (5,189 ft), and 01/19 (6,869 ft). Because of the relatively short length of its runways, DCA has short to medium-size transport aircraft operations. In this preliminary analysis, the gates are aggregated into 12 groups and it is assumed that several flights can occupy one gate area at the same time.

B. Flight Demand

Nominal Sequence for runway operations: Using random number generation techniques, the ready times (or nominal times) for runway operations are generated. A total of nine levels of hourly demand (ranging from 10 to 50 flights per hour) are generated. To consider randomness of generated data, ten data sets are generated for each level of hourly demand. In all test scenarios, the interval for any two consecutive runway operations (either takeoff or landing) follows a negative exponential distribution. The aircraft mix is another important factor influencing the capacity of an airport. For this study we use 0% heavy, 70% large and 30% small aircraft. The ratio of arriving and departing flights are set to 50:50. For computational simplicity, it is assumed that all flights use a single runway 01. Using nominal times for runway operations, the activation times for departing flights are computed by subtracting the sum of nominal taxiing, communication and some buffer times from nominal times. A specific gate is also assigned to each flight at the airport terminal.

First-Come-First-Serve (FCFS) with landing priority sequence for runway operations: In order to simulate and air traffic controller sequencing process, the so-called First-Come-First-Serve (FCFS) with landing priority sequencing strategy is applied. The main idea of this sequencing scheme is to adjust the nominal arrival sequence in such a way that arriving aircraft should not be delayed by departing flights. In other words, an arriving aircraft is delayed only to resolve a conflict with another arrival. The sequencing procedure is conducted in following way: The first step is to check if there are any conflicting arrivals. If any, the conflicting arrivals are separated by delaying the following arrival so that the minimum arrival separation is enforced. In the second step, if the inter-arrival time between two consecutive arriving aircraft is sufficient for one or more departures to be operated, then departing aircraft are scheduled. Otherwise, departing aircraft are to be delayed until a sufficient gap exists between two successive arrivals. In the simulation model, controllers use this sequence as a reference when they make a sequencing decision. Figure 7 shows an example of the flight schedules used in the model.

Activation times: Activation times are the times when flights are inserted into the simulation. The activation time for a departing flight is obtained by subtracting the sum of the nominal taxing time, communication time and certain amount of buffer time from its nominal time. The link and node data needed to define the taxiway topology are used to obtain nominal taxing times. Unlike departing flights, an activation time for an arriving flight is obtained by subtracting only communication time from its nominal time.

C. Time Durations for Communication Procedure

The time durations of the four steps in any communication activity provide a measure of the controller workload in the control tower. For simplicity, we assume that the time intervals for two communication systems are deterministic as shown in Table 2. Here, we further assume that the times required for sending messages are negligible under the data link system compared to the voice communication system.

2. Simulation

Each simulation is executed for a two-hour duration (i.e., 7200 seconds). This is done to secure enough time for all flights to finish their operations during the simulation duration. The system clock is set to advance by one second. This time interval, $dt$, is an important global setting associated with aircraft-following model. There is a tradeoff in selecting the size of $dt$. Smaller $dt$ provides more detailed results in aircraft behavior but requires more time and storage space in during the simulation.
3. Computational Results

In order to compare efficiency of two communication systems, average delays for the operating aircraft are computed by dividing the total delay by total number of flights. The total delay is defined as the sum of time differences between two completion times of the nominal (or unimpeded) operations and actual operations. In VTASIM the completion time of a nominal operation is the time duration that is needed for a flight to complete its operation impeded by no other flights. The total delay is computed by adding two types of delay effects: a) the runway delay and b) the taxiing delay. These delays are that are defined as follows:

Runway delay: Is the difference between the time when an aircraft is scheduled to start its runway operation without any delays and the time when the actual operation takes place.

Taxiing delay: The taxiing delay accrues whenever any flight’s taxiing speed is less than a nominal taxiing speed. The taxiing delay is estimated by the following equation.

\[
\text{Taxisiing delay} = \int_{0}^{\text{simulation time}} \left( \frac{\text{nominal speed} - \text{current speed}}{\text{nominal speed}} \right) dt.
\]

The resulting average delays obtained at different flight demands are shown in Figure 8. The result indicates that the delays using data link are considerably less than those using voice channel and the average delay can be reduced up to 35 % when CPDLC system is used.

CONCLUSION AND RECOMMENDATIONS

In this paper, we quantified the benefit of implementing a CPDLC system over conventional voice communication system. For the analysis, we applied a microscopic simulation model named VTASIM to the DCA airport at different levels of flight demands. The result of the analysis reveals a very promising future of CPDLC system that can reduce the average flight delay by 35%. The results of this analysis require further investigation and validation of the delay times assumed for both the voice and the data link systems. These times can be obtained using human-in-the-loop studies. Ryan (1992) suggested that in order to achieve two goals of minimizing delays and maximizing safety, the best combination of two communication systems would be 90% of data link and 10% of voice radio communication. Motivated by this comment, this study can be enhanced by considering mixed communication systems of data link and voice communication systems. Another possible enhancement to the simulation model would be the incorporation of stochastic behaviors for flights and controllers to better represent the variations of message length and service times.

REFERENCES:


List of Figure and Tables

Figure 1. CPDLC Message Display On-board the FAA Boeing 727.  
Figure 2. Boeing 757 with CPDLC Equipment.  
Figure 3. Flight Progress Strip.  
Figure 4. State Diagram for a Controller.  
Figure 5. Communication Process Initiated by the Controllers.  
Figure 6. State Diagram for Communication (Voice Channel).  
Figure 7. State Diagram for Communication (Data Link).  
Figure 8. Flowchart for the Overall Simulation Model Process.  
Figure 9. Comparison of Average Delays per Flight for Two Communication Systems.  

Table 1. The State Variables and Initial Values.  
Table 2. The Durations of Communication Activities.
Figure 1. CPDLC Message Display On-board the FAA Boeing 727 (Courtesy of FAA: http://ffp1.faa.gov/tools/tools_cpdlc.asp).
Figure 2. Boeing 757 with CPDLC Equipment.

(Courtesy of FAA)
Figure 3. Flight Progress Strip.
Figure 4. State Diagram for an Air Traffic Controller.
Figure 5. Communication Process Initiated by the Controllers.
Start Communication

Put this flight strip to progressing box.

Is controller busy?

No

Sending Request (1)

Waiting Command (2)

Receiving Command (3)

Sending Confirm. (4)

End Communication

Ready to comm.

Sending Confirm. (4)

Receiving Command (3)

Wait Contact from Controller

Received clearance?

No

(i.e., Delayed)

Yes

Put this flight strip to progressing box.

Start Communication

Sending Request (1)

Is controller busy?

No

Waiting Command (2)

Receiving Command (3)

Sending Confirm. (4)

End Communication

Ready to comm.

Sending Confirm. (4)

Receiving Command (3)

Wait Contact from Controller

Received clearance?

No

(i.e., Delayed)

Yes

Figure 6. State Diagram for Voice Channel Communications.

Figure 7. State Diagram for Data Link Communications.
Initialization
// read data
// generate network graph, controller, flights

sysTime = 0

Flight = 1st Flight in List

sysTime == nextMoveEventTime?

Yes

call checkNeedToCommunicate

needToComm == True &&
  sysTime == nextCommEventTime?

Yes

Communication

Movement

Last Flight in the flight list?

Yes

sysTime == simDuration?

Yes

End

No

Flight = nextFlight

sysTime = sysTime + dt

No

Figure 8. Flowchart of the VTASM Simulation Model.
Figure 9. Comparison of Average Delays per Flight for Two Communication Systems.
Table 1. The State Variables and their Initial Values.

<table>
<thead>
<tr>
<th>Entities</th>
<th>State Variable</th>
<th>Initial Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>controller</td>
<td>currState</td>
<td>standby</td>
</tr>
<tr>
<td></td>
<td>nextEventTime</td>
<td>simulationDuration</td>
</tr>
<tr>
<td>flight</td>
<td>currCommState</td>
<td>readyToCommunicate</td>
</tr>
<tr>
<td></td>
<td>nextCommEventTime</td>
<td>simulationDuration</td>
</tr>
<tr>
<td></td>
<td>currMoveState</td>
<td>parking/onFinal</td>
</tr>
<tr>
<td></td>
<td>nextMoveEventTime</td>
<td>scheduled time</td>
</tr>
<tr>
<td></td>
<td>Position</td>
<td>gate/runway threshold</td>
</tr>
<tr>
<td></td>
<td>Speed</td>
<td>0.0/final approach speed</td>
</tr>
<tr>
<td></td>
<td>Acceleration</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>needToComm</td>
<td>false</td>
</tr>
<tr>
<td></td>
<td>collisionChecked</td>
<td>false</td>
</tr>
<tr>
<td></td>
<td>Permission</td>
<td>undecided</td>
</tr>
</tbody>
</table>

*State for arrival/departure.*
Table 2 Duration Times for Communication Activity.

<table>
<thead>
<tr>
<th>Activities for pilot (Corresponding activities for controller)</th>
<th>Duration (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>voice communication system</td>
</tr>
<tr>
<td>Sending a request (Receiving a request)</td>
<td>4</td>
</tr>
<tr>
<td>Waiting a command (Judging the request)</td>
<td>3</td>
</tr>
<tr>
<td>Receiving a command (Sending a command)</td>
<td>4</td>
</tr>
<tr>
<td>Sending confirmation (Receiving confirmation)</td>
<td>3</td>
</tr>
</tbody>
</table>