Fault-Tolerance of E-Commerce Transaction Protocols

Tricia Yaw

Advisor: Elizabeth Borowsky

May 2003

BA Thesis

Motivation

My interest in e-commerce transaction protocols stemmed from numerous questions I had about the nature and reliability of online purchasing. I wanted to know how the major protocols worked and how unlikely or likely it was for a transaction to be lost in the void. Having knowledge of computers and networking, as well as being an internet shopper myself, I wondered how the unpredictability of the internet might affect the average user.

Background

E-commerce began in the early 1990s, as the internet grew. One of the primary business forces driving the growth of the internet was the potential for on-line shopping. To enable both the trust of users of business transactions as well as to guard against on-line scams, Visa and MasterCard teamed up to develop a secure method of transfer. Secure Electronic Transmission (SET) was created to allow safe credit card transactions over the internet. As the late 1990s grew into the dot-com boom, different protocols were developed for varying purposes. Digicash created Ecash, in the expectation that users would want an anonymous means for purchasing products, and Paypal came into existence to enable "regular" people a means of purchasing items from each other.

Today, Visa can be used at most online stores, and these stores encompass nearly any item that you could potentially buy. This allows countless people to shop conveniently from the privacy of their homes. Also, numerous companies allow bills to be paid online, often automatically from a credit card.

Ecash, an anonymous electronic payment method developed by DigiCash, involved purchasing electronic payment certificates that could be verified by a bank and ideally used in every online market. To withdraw some electronic "cash", a customer contacts Digicash wanting to purchase an amount of money. Digicash takes a credit card payment for this and sends one hundred electronic envelopes to the bank all containing the same amount of money. The bank opens ninety-nine of the envelopes and assumes that the hundredth holds the same amount. Then the bank marks this envelope as digital cash and sends it back to Digicash. Digicash forwards it on to the customer. The customer can use this Ecash himself or he can give it to someone else to use, and merchants would accept it because it has been certified by a bank. This idea did not catch on, and while it is still being developed, it has lost its drive. As it turns out, people are not as worried about on-line merchants tracking their purchases as many had expected initially.

Paypal, which will be discussed in detail later on, became an answer to the rising number of private citizens buying and selling on the internet. With the fees and hassle, an individual will often opt to not accept credit cards. While this makes sense, it created a major problem. An individual needed a way to receive money from another individual faster than by mail. Paypal acts as an online bank which allows anyone with a bank account in a traditional bank to send and receive funds electronically. The most common usage for Paypal is in completing transactions from internet auction sites such as eBay. It is also essentially a free wire transfer between almost any two people.

Visa Protocol

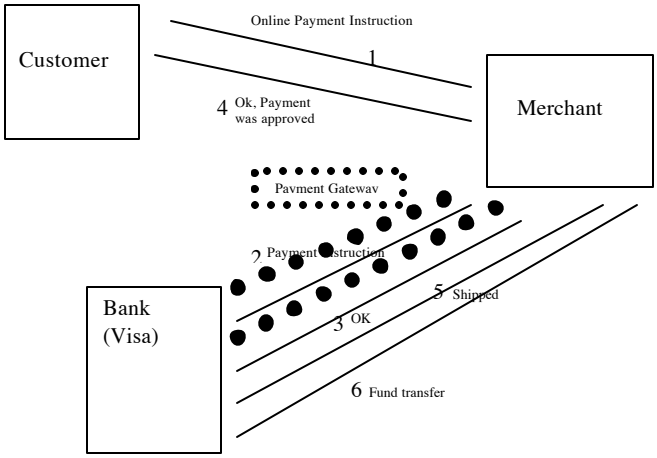


Diagram 1

Diagram 1 demonstrates one complete transaction using Visa through an online merchant. The customer begins the process by sending information to the merchant regarding the desired purchase, Message 1 above. As shown in Diagram 1, the merchant forwards the credit card information and personal information on to the bank. The bank verifies that the account is valid and has enough credit for the purchase, and sends either an approval or denial back to the merchant, specifically Message 3. This confirmation is forwarded on to the customer. If the customer receives a denial, then the transaction ends, and the customer has the option of reinitiating the transaction. If approved, the merchant sends the product or service, and the customer just waits for the delivery. Once the merchant has sent out the product, he sends Message 5, a message to the bank confirming that the transaction has completed. At this time the bank will transfer the funds to the merchant.

Paypal Protocol

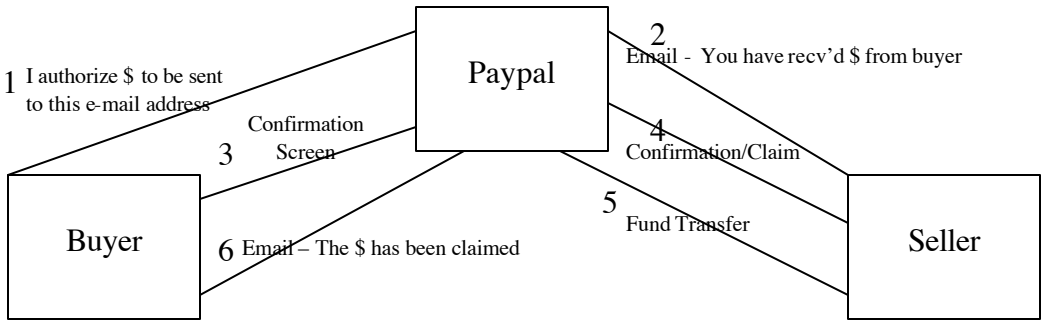


Diagram 2

Diagram 2 illustrates the protocol used by Paypal. In this case, there are three participants; however two of them are individuals with only one player being a computer. The buyer initiates the transaction by sending Message 1 to Paypal indicating that he wishes an amount of money be transferred from his personal account to that of a seller by the seller's e-mail address. Paypal sends an e-mail to the seller stating that the seller has received a money transfer from the buyer, which must be claimed. Once the e-mail goes out to the seller, the buyer receives confirmation, Message 3, of the contact from Paypal. Once the seller confirms, namely Message 4, the money is placed into his account. Paypal then sends Message 6, an e-mail to the buyer that the money has been claimed.

Visa Simulations

In Visa there were five key points that needed simulation and examination. They were corruption between customer to merchant, corruption and dropped messages between merchant to bank, and corruption and dropped messages between bank to merchant. In practice the transaction line between merchant and bank and between bank and merchant are essentially the same issues and can be addressed as one.

If either the merchant or the bank receives corrupted data, it will send a message requesting the sender retransmit the information. This will continue until corrected or until the receiving party times out. Since the customer can only receive corruption when receiving the confirmation, we do not need to have him request a resend, as he should be able to do check the status of his transaction at any given time through the merchant.

Dropped messages are more complex, as the receiving end does not have any record of their existence. The merchant has a timeout function while waiting for a response from the bank, if he does not receive a response before he times out, he will resend his data. The bank does the same once contacted by the merchant. A dropped message from the customer cannot be traced and must be checked on by the customer.

Paypal Simulations

Paypal had only two cases where either corruption or dropped messages impacted the reliability of the protocol. If Paypal receives a corrupted message from either the buyer or the seller, it will send a request for the sender to retransmit the data. These requests will continue until Paypal receives a non-corrupted message.

The reason so little could be simulated in Paypal concerns the two individuals. While Visa has two stages which are computers which must receive things in a certain order, in a certain way, Paypal has a free thinking individual on each end. A person can look to see if they have received messages without prompt. A computer needs to know that he is expecting in order to check. This made Paypal more confusing and more difficult to simulate.

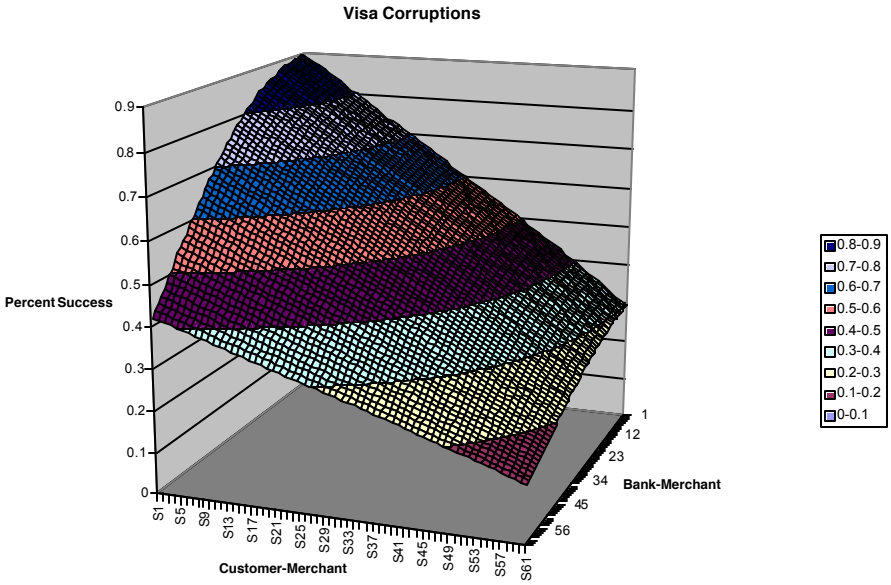
Testing

For both Paypal and Visa, I ran tests of a size 500,000 twice each. The two different variables for each were corruption rate and dropped message rate. Thus, I basically ran four tests of size 500,000. For each of the corruption tests, I fixed the dropped message rates to be five percent. Then I created two for loops, one nested in the other. The nested for loop counted from zero to sixty percent, for the transactions between the seller and bank for Paypal and the merchant and bank for Visa. The outer for loop also incremented from zero to sixty percent, this time for the transactions between Visa's customer and bank and Paypal's buyer and bank. This ensured that every combination of percentages between zero and sixty would be covered in the test. The same design was used for the dropped message percentages.

Results - Visa

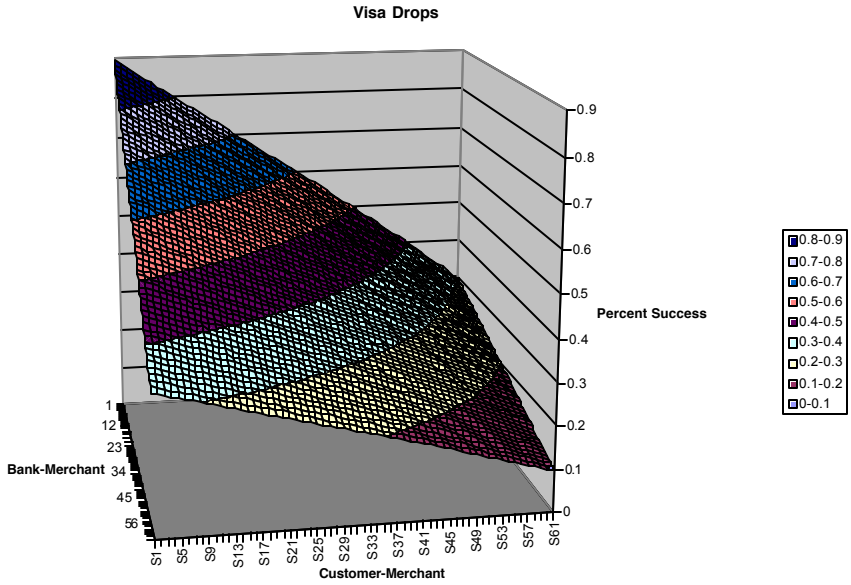
Figure 1 represents Visa Corruptions, which is fairly predictable, as each corruption rate increases, the stability of the protocol decreases. As can be seen in the graph, the success rate when the corruption rates are equal is essentially a straight line. However, as either Bank-Merchant or Customer-Merchant corruption rates increase, the graph begins to drop off.

Because in Visa, two of the four connections are easily corrected for drops, and two must be corrected manually by a human, the graph is quite consistent throughout. In Figure 2, there is a slight decrease in success for the protocol when either the Bank or the Customer connection increases in drop rates. Even at maximum drop levels, the protocol is still reliable approximately one in ten times, which is not great, but is at least still succeeding.



In this diagram, the values of the Percent Success axis range from 0-90%. The values of the Customer-Merchant axis range from 0-61. The values of the Bank-Merchant axis range from 0-61.

Figure 1



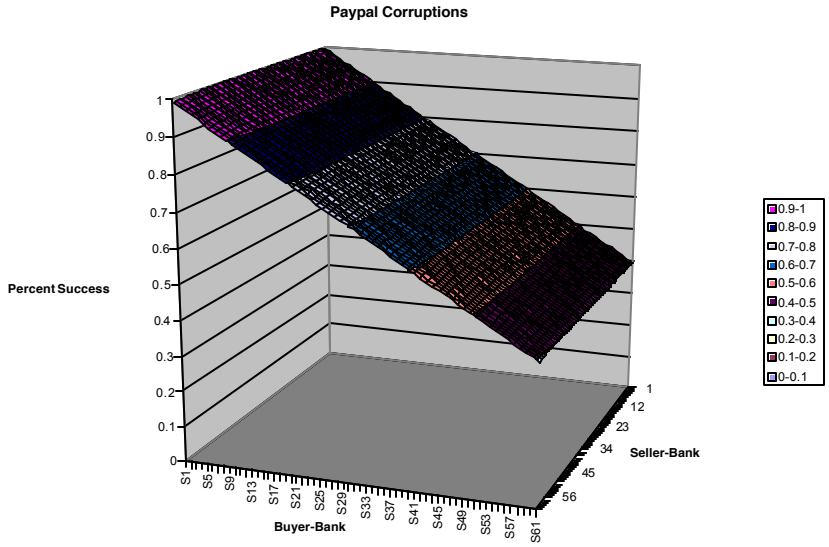
In this diagram, the values of the Percent Success axis range from 0-90%. The values of the Customer-Merchant axis range from 0-61. The values of the Bank-Merchant axis range from 0-61.

Figure 2

Results - Paypal

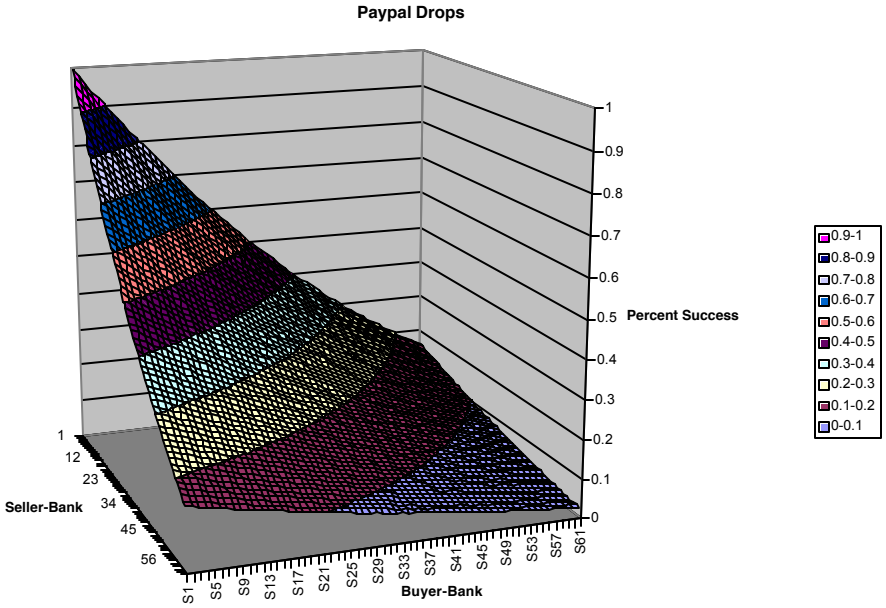
As Figure 3 shows, as the Bank's connection with either the Seller or the Buyer begins to corrupt, it does not affect the success rate as much as with the Visa protocol. This is because there are two human factors in this protocol, which do not require the Bank to resend data if corrupted, as they can access the information from the Bank themselves. This allows the reliability of the success of Paypal corruptions to be quite good.

Dropped Message rates in Paypal cause problems though, as seen in Figure 4. Because Paypal's Bank does not necessarily know if a Buyer contacts it, this is one problem. Another factor is that the Seller may not know if he is expecting to be contacted by the Bank. A huge issue is also that the Bank does not expect the Seller to respond, so it does not contact the Seller for a reminder. With these alone, nearly half of the transaction protocol is exceptionally vulnerable to dropped messages. But, as mentioned before, these can be accounted for by the human involvement on each end, making the graph in Figure 4 slightly deceiving.



In this diagram, the values of the Percent Success axis range from 0-100%. The values of the Buyer-Bank axis range from 0-61. The values of the Seller-Bank axis range from 0-61.

Figure 3



In this diagram, the values of the Percent Success axis range from 0-100%. The values of the Seller-Bank axis range from 0-61. The values of the Buyer-Bank axis range from 0-61.

Figure 4

Possible Future Work

Had I more time, I would have liked to examine more variants in the protocols. For instance, I was not able to simulate load capacity for the servers. I wanted to have three programs per protocol to simulate the actual usage of the bank or the merchant. In the example for Visa, a program for the customer would generate increasing numbers of customers sending requests to the merchant, looking for how the merchant would recover in less than ideal situations. Eventually I would like to also simulate this while increasing the number of merchants to test the load of the bank.

Though there is no way to adequately graphically represent the data, I would have liked to have the corruption and dropped message rates integrated such that they affect each other. That would require 5 axes, and it would be difficult to analyze the results. But this would also be a more accurate indicator of a protocols fault tolerance.

Conclusions

While both faired well in my tests, I believe that Paypal came out on top as far as fault-tolerance was concerned. That may be mostly due to the fact that there are two humans involved, and humans are quite capable of recovery from drops or corruptions. As far as the protocol itself, Visa does the most error correction simply because there are two machines that must keep functioning. The security of Paypal is faulty at best because there is no method to verify that the actual recipient is indeed the intended recipient. Paypal If security is not of the utmost importance, then Paypal is the method of transfer I would use more frequently.