OFFLINE ARCHITECTURE FOR REAL-TIME BETTING

Panu Hämäläinen1, Marko Hännikäinen1, Timo D. Hämäläinen1, and Riku Soininen2

1Tampere University of Technology, Institute of Digital and Computer Systems
P.O. Box 553, FIN-33101 Tampere, Finland
panu.hamalainen@tut.fi, marko.hannikainen@tut.fi, timo.d.hamalainen@tut.fi

2Oy Veikkaus Ab
Hämeenpuisto 44 C, FIN-33009 Tampere, Finland
riku.soininen@veikkaus.fi

ABSTRACT

Traditional betting systems do not enable betting in real-time during an event and require decisions and preparations beforehand. In this paper, a novel architecture for real-time betting is presented. It disposes of the up-front effort and enables frequent bet announcements and placements during an ongoing event. The novel situation is achieved by broadcasting announcements, time-stamping and storing the placements locally, and collecting them after the event has been finished. While solving the processing problems, the architecture requires reliable cryptographic and physical protection. Currently, e.g. DVB and LAN technologies offer potential platforms for providing the service. The implemented LAN demonstrator has shown that user interfaces have to be simple and bets should not be announced too often. It has also shown that real-time operation makes betting more inspiring.

1. INTRODUCTION

Betting has traditionally been organized manually by placing bets in front of a retail operator. However, it has also found its electronic forms abreast of the rapid development of the information technology. A number of organizers, e.g., the governmental lottery monopoly Oy Veikkaus Ab [7] in Finland offer so-called online betting services on the Internet.

The benefits of electronic betting are indisputable for both the organizer and the customer. In addition to reduced setup time and effort, it is possible to place bets remotely and closer to the beginning of the target event as well as to collect the winnings automatically.

In general, there is plenty of time for placing bets and betting has to take place quite a long time before the target event begins. The bet incidents are long lasting (e.g. a horse race) and their existence as well as start and end times are accurately known in advance.

An example of an incident in a soccer match that cannot be announced as a bet target could be “this penalty kick will result in a goal”. The counter or the communication network and server would become overloaded. A drawback in the existing systems is also that a person placing a bet loses the excitement in case the result of the bet incident becomes obvious already in the beginning. Likewise, if the bet incident is very short (“team Finland will score during the first power play”) customers probably choose not to participate. In addition, customers must decide before the event starts whether or not to place bets, on which incidents, and how much money they like to spend.

Although the electronic systems have made betting convenient, they have not changed the fundamental nature of betting. The targets available in electronic systems are also suitable for manual systems. Just the means for placing bets and collecting winnings have evolved.

To improve the characteristics of betting itself, a Real-Time Betting (RTB) architecture has been developed at Tampere University of Technology (TUT). As opposed to the online systems as well as the ones presented, e.g., in [3], [8], [9], and [12], the proposed architecture does not require communication between betting terminals or towards a server during bet placements. The novel situation is achieved by utilizing broadcast downlink transmissions and offline uplink terminals. The architecture scales well and sets only reasonable processing requirements to its components. A RTB demonstrator for LAN environment has been developed for prototyping purposes.

2. REAL-TIME BETTING

In this paper, the term event refers to a whole betting target (e.g. a soccer match) which comprises of several incidents (e.g. a penalty kick) on which bets may be placed. A session includes all the announced bets between the session’s start and end times during an event.

RTB is defined as an activity in which an operator continuously and frequently announces targets during an event and customers place a number of small bets on these unexpected incidents. After the outcome of a bet incident is known, customers immediately get their winnings back and may invest them on new incidents during the same event. The goal of RTB is to make betting an interactive session with frequent bets and immediate feedback. RTB enables new types of bet incidents as well as disposers of the time-consuming effort required beforehand in the traditional systems.

For the practical applicability of RTB there are clear requirements set on bet incidents. They have to be single, distinctly separable, and their outcomes unarguable. When the event is fairly well predictable and sequential, it is possible to prepare suitable bets as templates in advance. This makes it easier for an operator (a person announcing bets) to react when potential incidents occur. For example, ski jumping is an event with suitable characteristics.
Another key feature in RTB is fairness among customers. Each customer should get his or her bets placed, independent of how close to the deadline the operation is initiated. This guarantees equal possibilities for gathering as much information as possible from the target event before making a decision.

Fulfilling the requirement is also an important part in creating the excitement of RTB. When the number of customers is small, RTB can be implemented with existing online systems. The limit for customers is set by server processing capabilities and network bandwidth. Most participants probably delay their bets near the deadline, especially if the rates are dynamically adjusted. Hence, preparing for high peak traffic is required.

Particularly, the service cannot fulfill the fairness requirement on the Internet, since customers with better network connections are favored. A better implementation consists of distributed, trusted time-stamping devices [10]. They communicate with the betting terminals appending unchangeable timing information to their data. At a suitable time, the betting server retrieves the data and computes the results.

For example, the stamping servers by Symmetricon [6] are capable of producing 50-125 secure time stamps per second, which implies a need for a variety of devices for large-scale RTB. In order to allocate the bandwidth and processing power evenly, the distribution of the devices has to be well-designed and the location of the terminals available beforehand. This requirement makes the implementation less flexible and unsuitable for wireless networks supporting roaming. Updating the credits in real-time is also problematic.

For the betting organizer the most important security aspect relates to timing. The server should not accept the bet placements after the deadline. It should also identify the falsified attempts to do so. In addition, the server must reliably authenticate and authorize customers, preventing them from appearing with false identities or access rights. On the other hand, for a customer security is close to fairness and availability. There should not appear a situation, in which a customer loses valid bets because of networking or processing problems. The implementation must also guarantee that a malicious party cannot tamper with transmitted bet placements or announcements making a customer lose.

3. OFFLINE RTB ARCHITECTURE

In order to overcome the real-time processing problems but still maintain the service intense, a new approach has been adopted at TUT. The novel architecture is referred to as offline RTB and it enables electronic betting in short time cycles. Regardless of the number of participants and bet announcement frequencies, the offline RTB only sets reasonable performance requirements to its components.

The offline RTB architecture is presented in Figure 1. It includes a server and a bet operator in the organizer premises, a network operator, a (separate) broadcast operator, and customer terminals. The broadcast link conveys bet announcements while the two-way link takes care of registrations and bet record collections. The term offline refers to the bet placement phase in which the terminals are receiving broadcast downlink data but sending nothing upstream. Betting in the offline RTB comprises of three separate phases: registration, betting session, and bet record collection. Figure 2 illustrates the high-level message exchange graph for the phases.

In the registration phase customers browse through upcoming events and choose the ones they wish to participate. Dedicated playing accounts may be provided or the money may be reserved through electronic bank transactions as in current online systems. The phase exploits the secured two-way data connection. When the registration is confirmed the time between the terminal and the server is synchronized and parameters for ensuring the security in the betting session are exchanged. The security-related information includes the public key of the server, timing information, and parameters for storing bet records. After registration the two-way link is closed.

The server notifies about the beginning of the target event with a session start broadcast packet. As a suitable incident emerges, the bet operator releases a bet in the server. The server broadcast a start bet packet containing an incident identifier, textual target information, rates, and result alternatives. Just before the outcome of the incident is settled the operator sends a stop bet packet. This contains information about a time instant after which placing a bet for the current incident is not allowed. Already the existence of the closing packet at a terminal disables placing more bets (placement 3 in Figure 2) but, in addition, the terminal checks whether a customer placed a bet before the time instant (placement 1). This guard period guarantees that the delay of the broadcast transmission or slow reaction by the operator do not allow placing bets after the outcome of the target incident is already known (placement 2). It also prevents a malicious party benefiting from transmitting a complete, recorded broadcast session data. If the tests pass, the terminal locally accepts, time-stamps, and stores records for the placed bets.

When the result of the incident is clear, the terminal receives an incident result packet, compares it to the stored
records, and informs the customer of the outcome. The terminal immediately adds the winnings to the available credit. However, the outcome calculations are only preliminary; the results are definitive only after the server has checked their validity after the session end. The betting continues similarly until a session end packet is transmitted.

A while after the betting session has finished the server requests the terminals to send their bet records back, or it has given the terminals unique transmission times earlier in the registration. This phase again utilizes the two-way connection. After receiving the bet records the server checks that the data has been correctly stored and the time-stamps are valid and computes the results. It is important that the server receives a record for every bet announcement, whether a customer placed a bet or not. Otherwise it is possible that a customer tries to cheat by not transmitting the placements that were not winning.

### 3.1. Security Challenges

While solving the problems in fairness and availability, the offline solution adds new constraints especially for security. Reliable methods are needed to locally guarantee that the bets are placed before outcomes are known and stored records have not changed afterwards. Since the whole operation environment of the terminals must be considered hostile, it is important that the customers or any other external party do not have any control over or access to the security procedures. In addition, real-time processing implies a need for efficient implementations.

For authenticity and privacy, the two-way data connection implements reliable authentication and encryption. On the broadcast side, each packet contains security information, which undeniably binds the packets to the betting server. Digital signatures [1] are suitable for the purpose. It is also necessary to ensure that the server can reliably identify the customers and that they cannot deny placing collected bets afterwards. Authenticating the customers at the terminal when logging on and at the server in the registration and bet record collection phases make this possible. The bet records may also include digital signatures for further enabling non-repudiation.

Since there is no traffic upstream during betting, keeping terminals active and synchronized with the server is important. In order to maintain the synchronization after registration, a terminal contains a watchdog timer that stays alive with constant broadcast packets (beacon packets in Figure 2) from the broadcast operator or alternately from a separate source. The server defines the beacon interval for a terminal in the registration phase.

For further security, the server may also define what actually constitutes a beacon. For example, the beacon can be a certain combination of bits in a transmission frame. The server may also alter the construction by secretly defining a new one during a betting session. The watchdog prevents from tampering with the broadcast traffic, i.e., delaying parts of it to enable tampering with the recorded information about the target event itself through some other media. The local time stamping results in that the oscillators and clocks of terminals have to be trusted and it must be impossible to tamper with them. Using the same oscillator for time and synchronization to the communication medium is one possibility; changing the oscillation speed disables also the communication. In addition, at least the encryption keys for storing betting data, and preferable the data storage itself, as well as the watchdog parameters have to be physically protected. Requiring the usage of an authentic and registered terminal assures that a malicious party cannot construct a device for falsely benefiting from RTB. Hardware, especially integrated circuit and smart card, implementations have potential for offering adequate physical security.

The server may also randomly request the terminals to transmit some summarizing data about their storage, program codes, and time concept. The offline RTB requires agreed politics and rules, which the organizer applies when exceptional situations occur. For example, the rules have to define the actions the organizer takes if, for some reason, the server cannot collect the bet records from a terminal. Moreover, the organizer should keep a log on the winnings so that if some customers start succeeding suspiciously often, the authenticity of their activities can be checked. In addition, limiting the sizes of the stakes makes the service less appealing for break attempts with financial goals.

### 3.2. Technologies

There are several alternatives for offline RTB implementation platforms. The emerging Digital Video Broadcasting (DVB) technology offers a potential solution for the both links. The video broadcast transmission can include bet announcements while registrations and bet record collections can utilize the available return channel. It is convenient to use the same terminal for following the event and placing bets.

Another solution for advanced multimedia terminals as well as small hand-held devices can utilize Digital Audio Broadcasting (DAB) technology. The customers may follow the event via DAB transmissions or some other media (e.g. TV broadcast or “on the spot”). However, in addition to a DAB receiver, another network connection for registrations and bet record collections is required.

Large-scale betting with existing technologies can use GSM/GPRS for the service. In that case, the customers obtain the information about the target event itself through some other medium. Again registrations and bet record collections use two-way data calls as bet announcements utilize broadcast traffic (e.g. short messages). Similarly to DAB, this alternative is suitable for small, portable terminals with limited processing capabilities since there is no need for video transmission reception and decoding. Unfortunately, the delays in these systems may decrease their applicability for RTB.
As supporting the both broadcast and two-way traffic, WLAN technology provides a suitable solution for localized services. The number of access points just has to be large enough to cover all the employed betting terminals. The wireless solution is convenient, e.g., in a stadium allowing customers to participate without tying them to a fixed spot. Another wireless technology for local betting services, e.g., in box seats is Bluetooth.

4. OFFLINE RTB DEMONSTRATOR

An offline RTB demonstrator has been implemented using Specification and Description Language (SDL) [5]. SDL allows implementing entities on different abstraction levels and connecting them together to form a single executable system for design verifications and simulations. In addition, the design tools enable generating application C source code directly from the SDL descriptions. The created demonstrator does not limit the number of terminals accessing the service and supports several operators and parallel events as well as announced bets.

The demonstrator has been developed for LAN/WLAN environment. It is suitable for localized services, e.g., in a building or a stadium. The setup for ad-hoc WLAN is presented in Figure 3. The server application and operator interfaces are separated so that the laptop/PC containing the server software can be located in a place with the best network coverage and physical security and operator laptop in a more suitable place for following the target event. The terminal software is available for both PCs and Compaq iPAQ Pocket PCs [4]. The system utilizes UDP for broadcasting and TCP for the two-way connections.

In order to activate a bet the operator utilizes predefined bet templates, which can be used as such or the texts and rates can be dynamically adjusted. Thus, the only compulsory task for the operator is to define the start and end times as well as the result. In addition to bets, the operator manages user accounts, bet templates, and receives real-time information about the ongoing sessions and registered customers. The customers can participate various events simultaneously and define the sums reserved for events and default stakes.

Suitable user interfaces for RTB have also been evaluated using the demonstrator. The tests have shown that the information must be very concise, because the time for placing a bet is strictly limited. In addition, otherwise betting disturbs too much following the event. The target incident should only have two or three outcome alternatives. Even though a clear and simple user interface makes betting convenient and does not affect following the event, the operator should not make announcements too often.

5. CONCLUSIONS

Advancing mobile networking technologies and their strengthening security enable realizing numerous new services, such as RTB. RTB changes the nature of betting from a static, slow, long lasting procedure into an intense, interactive session. A different approach from the online betting systems has been adopted, referred to as offline RTB. It makes the service scalable and enables betting in short time cycles while keeping processing requirements low.

In the offline RTB bet placements are made locally without interaction with the RTB server. In addition, when two-way connections are required, their existence is distributed into a larger time span. A drawback in the developed architecture is that it makes the security design challenging. The whole operation environment of the betting terminals has to be considered hostile. Thus, reliable cryptography and time synchronization as well as physical security are needed.

The constructed RTB demonstration enables prototyping as well as evaluating suitable network technologies and user interfaces in practice. The tests have shown that RTB requires simple and clear user interfaces. In addition, it must be possible to make all selections with a single keystroke and the bets should have only a few possible outcomes. Even though the offline RTB is in the development phase, it has been shown that the service is inspiring and entertaining.

6. REFERENCES