CARBONITE | WEBROOT
an opentext company | an opentext company

# Replication decoded

## How different replication technologies affect data protection

Periodic versus real-time replication is no longer an either-or proposition. With today's high level of system complexity, it's necessary to deploy both. When considering how to deploy protection for a particular dataset, first decide if the goal of protection is for reactive recovery or proactive productivity. Simply stated, periodic replication is best for recovery while real-time replication is best for productivity.

### Periodic replication

Periodic or snapshot replication captures a point in time that organizations wish to preserve for various reasons, such as archiving, backup and disaster recovery. It lets businesses create a historical record consisting of incremental, time-specific datasets, usually on a daily and monthly schedule. By capturing data at a specific point in time, organizations can preserve inactive data, mine for business intelligence and recover clean versions of compromised data.

### Real-time replication

Real-time or continuous replication actively mirrors server workloads, capturing changes at the byte level and synchronization parity between the source and the secondary target. It allows organizations to ensure high availability of critical servers and applications. This is the optimal choice when the urgency of data is high and the consequences of downtime are severe to catastrophic. Real-time replication also allows organizations to migrate server workloads for periodic maintenance, software patches and technology upgrades.

### Replication best practices

The best data protection strategy today blends the different types of replication to achieve strategic objectives within an overarching data protection plan. That means periodic replication for all data across all systems and real-time replication for mission-critical, time-sensitive data.

### Carbonite real-time replication

Both Carbonite® Availability and Carbonite® Migrate use real-time replication. Our replication technology is engineered to quickly, safely and efficiently identify and transfer byte-level changes, and ensure they are written correctly on the target, all while using minimal CPU or bandwidth.

### Replication workflow

Our real-time replication engine starts by performing a file-level comparison of the existing data between source and target. Any file segments that are identical are skipped. Those with differences are copied from source to target. There's also the option to pre-seed the target by loading a flat-file copy of the data ahead of time. Replication begins and continues throughout the mirror process. If there's a disruption between the two servers (short of a failover event), the replication engine will automatically re-mirror using this same methodology with no administrative intervention necessary.

### Checksum

A series of checks ensure data is perfectly mirrored between source and target. The software first ensures the operation was received without errors and in the correct file-system write order based on the source that sent it. This process happens continuously. While the software identifies new input/output (I/O), buffers handle bottlenecks and an acknowledgement queue provides a re-transmission mechanism to handle in-line errors caused by TCP anomalies or other networking issues.

### Bit-perfect

Carbonite byte-level replication eliminates bursts of potentially large volumes of data across the network. This makes it ideal for environments with limited bandwidth and distributed networks. It's also application-agnostic, so it works with any application or database management system.