The given recurrence relation is of the form $T(n) = 2T(\frac{n}{2}) + 2^n$, where n is a power of 2. To solve this recurrence relation, we can use the substitution method.

Let's assume that $T(n) = S(n) \cdot 2^n$, where $S(n)$ is some function. Substituting this into the original recurrence relation, we get:

$S(n) \cdot 2^n = 2 (S(\frac{n}{2}) \cdot 2^(\frac{n}{2})) + 2^n$

Dividing both sides by 2^n, we get:

$S(n) = 2S(\frac{n}{2}) + 1$

This is a much simpler recurrence relation, and we can solve it by substitution. Let's assume that $S(n) = 2^k$. Substituting this into the above recurrence relation, we get:

$2^k = 2 \cdot 2^{k-1} + 1$

Expanding 2^k and 2^{k-1}, we get:

$2^k = 2^{k+1} - 2$

This equation holds for all positive integers k. Hence, we can conclude that $S(n) = 2^{\log_2 n}$, where $\log_2$ is the logarithm to base 2.

Finally, substituting this back into $T(n) = S(n) \cdot 2^n$, we get:

$T(n) = 2^{\log_2 n} \cdot 2^n = 2^{n + \log_2 n}$

So, the time complexity of the given recurrence relation is $O(2^n)$.